JOURNAL OF ENGINEERING DESIGN
https://doi.org/10.1080/09544828.2026.2639928

Taylor & Francis
Taylor & Francis Group

Codified collaboration: reinforcement learning with verifiable feedback as a mechanism for human–AI co-creation in generative intelligent design

Meno-Said Haddad and Arthur Seibel

Leuphana University Lueneburg, Lüneburg, Germany

ABSTRACT

Generative intelligent design systems typically remain static: they generate plausible artifacts but do not adapt through use or internalise expert design knowledge in a principled way. To address this limitation, we propose reinforcement learning from verifiable feedback as a framework for codified human–AI collaboration, in which domain expertise is formalised as algorithmic verifiers that evaluate generated designs against explicit structural and logical rules and convert rule compliance into reinforcement signals. We instantiate RLVF on the engineering task of functional decomposition, where designs are represented as typed function–flow graphs governed by verifiable structural constraints. Using group relative policy optimisation, a Llama-3.1-8B model is trained exclusively from verifier feedback without labelled output data. Across reinforcement learning cycles, correct output formatting improves from 18% to 100%, fully connected functional structures improve from 10% to 100%, and error-free decompositions improve from 4% to 100%, exceeding a supervised fine-tuned baseline trained on 257 labelled examples. Human evaluation shows that RLVF-trained models achieve higher perceived logical coherence than supervised models, while exhibiting reduced structural diversity. These results demonstrate that verifiable feedback can replace human annotation in rule-governed design tasks and enable correctness-grounded generative design systems based on codified expert knowledge.

ARTICLE HISTORY
Received 30 October 2025
Accepted 27 February 2026

KEYWORDS
Human–AI collaboration; generative intelligent design; reinforcement learning from verifiable feedback (RLVF); codified expertise; functional decomposition

1. Introduction

Generative artificial intelligence (AI) represents a profound socio-technical transformation in how humans and machines create, design, and make decisions. Moving beyond traditional analytic AI, generative systems can produce novel, meaningful content ranging from text and images to design concepts by learning patterns of reasoning and creativity from vast data corpora (Feuerriegel et al. 2024). These technologies, typified by large language models (LLMs) and other generative transformers, have turned AI from a passive analytical tool into an active collaborator in creative and design activities (Zhu and Luo 2022). In this new paradigm, humans and AI systems co-create and jointly explore conceptual

CONTACT Meno-Said Haddad meno-said.haddad@leuphana.de Leuphana University Lüneburg, Universitätsallee 1, Lüneburg 21335, Germany

spaces that extend human imagination while introducing new challenges of authorship, originality, ethics, and trust (Verganti, Vendraminelli, and Iansiti 2020).

Within this broader shift, engineering design is experiencing a parallel evolution: from traditional paradigms, such as knowledge-based engineering and intelligent computer-aided design, to generative intelligent design, where AI integrates domain-specific data, knowledge, and computation to autonomously generate and evaluate design alternatives (Bordas et al. 2024; Regenwetter, Nobari, and Ahmed 2022). This transition promises a step change in efficiency, exploration, and creative potential. Yet, despite these advances, most existing generative intelligent design systems remain static. They generate results by sampling from past data distributions but fail to adapt through use or learn from designer interaction. Crucially, they lack a mechanism to *codify expert design knowledge* and transform it into a learning signal that enables sustained human–AI collaboration. As design theorists note (Bordas et al. 2024), such systems practice 'design by activation,' reproducing learned patterns rather than exhibiting the combinatorial or emergent creativity that defines breakthrough innovation.

Empirical studies of design practice reinforce this limitation (Saadi and Yang 2023): even in advanced generative workflows, human expertise remains indispensable. Designers continue to define problem frames, manipulate constraints, and interpret outcomes, guided by domain intuition and value judgment. Rather than replacing human creativity, generative tools reposition it toward the curation and interpretation of machine-generated ideas. This evolving relationship echoes J.C.R. Licklider's early vision of *man–computer symbiosis*, where humans contribute creativity, goals, and critical evaluation, while machines provide computational reasoning and memory (Taylor 1960).

Achieving this vision, however, requires moving beyond one-way automation toward adaptive, two-way collaboration. Reinforcement learning (RL) provides a natural foundation for such adaptivity. Prior studies in human–AI collaboration demonstrate that RL enables agents to align with human intentions, coordinate effectively, and improve dynamically through feedback (Li et al. 2025). When evaluative signals are available during interaction, RL agents can learn faster, perform more reliably, and reduce cognitive load for their human counterparts (Islam et al. 2025). These findings suggest that RL is not merely an optimisation method but a mechanism for building interactive and trustworthy design intelligence.

Recent theoretical and empirical advances in reinforcement learning further reinforce this direction. Verifiable or binary feedback can serve as a stable foundation for adaptive optimisation, offering theoretical convergence guarantees and improving structured reasoning and problem-solving performance (Mroueh 2025). Complementary research on interpretable and symbolic feedback mechanisms demonstrates how human-readable rule systems can encode domain knowledge in transparent, modifiable forms, bridging the gap between human expertise and machine learning (Bougie, Onishi, and Tsuruoka 2024). Moreover, large-scale experiments with procedurally verifiable learning environments reveal that such structured feedback enables scalable and reproducible learning without the need for static, human-labelled datasets (Stojanovski et al. 2025).

Building on these developments, we argue that effective human–AI collaboration in generative intelligent design requires verifiable feedback grounded in explicit domain rules, through which expert knowledge can be made operational for learning. In engineering design, usefulness is constrained by correctness: generated artifacts must satisfy

physical, functional, and logical constraints to be meaningful. Violations of such constraints render outputs invalid, limiting trust and applicability in technical workflows.

To address this challenge, we introduce reinforcement learning from verifiable feedback (RLVF) as a framework for codified human–AI collaboration in generative intelligent design. RLVF embeds human expertise into algorithmic verifiers that evaluate AI-generated outputs against formalised design rules and transform these evaluations into reinforcement signals, enabling adaptive generative design systems. This enables adaptive learning while maintaining determinism, reproducibility, and auditability.

We demonstrate RLVF on the foundational task of functional decomposition (FD), where technical systems are represented as typed function–flow structures governed by explicit structural constraints such as admissible input–output patterns, network connectivity, and syntactic validity. These properties make FD particularly suitable for verifiable learning, as core correctness conditions can be checked automatically without subjective annotation.

Methodologically, RLVF is instantiated using a reinforcement learning procedure based on group relative policy optimisation (GRPO) (Shao et al. 2024), yielding an iterative loop of generation, verification, and policy improvement driven exclusively by verifiable signals. In addition to automated verification metrics, we examine how verifiable reinforcement influences broader generative behaviour, including its effects on structure diversity and creativity.

This paper makes four contributions:

1. It formalises reinforcement learning from verifiable feedback as a mechanism for codified human–AI collaboration in generative intelligent design.
2. It instantiates this framework on the task of functional decomposition using explicit structural verifiers and a GRPO-based learning procedure.
3. It demonstrates empirically that verifiable reinforcement substantially improves structural validity compared with supervised fine-tuning, without requiring additional labelled data.
4. It analyzes the broader generative effects of verifiable reinforcement through population-level mode-collapse analysis and controlled human evaluation.

Together, these contributions establish verifiable feedback as a principled foundation for adaptive and trustworthy generative design systems. By integrating rule-based verification directly into the learning loop, the proposed approach reframes human–AI collaboration as an explicitly codified and inspectable process.

2. Background

2.1. Human–AI collaboration with reinforcement learning

Early work on man–computer symbiosis proposed that humans and computers should function as complementary partners rather than substitutes. In this view, people define goals, hypotheses, and evaluations, while machines handle mechanical and data-intensive tasks that support real-time decision-making and creative insight (Taylor 1960). This vision foreshadowed today's efforts to create interactive AI systems that augment, rather than replace, human capabilities.

Reinforcement learning models sequential decision-making under uncertainty and captures the essence of adaptive behaviour seen in humans and animals. By learning from delayed consequences and balancing exploration with exploitation, RL agents can autonomously improve performance over time. These characteristics make RL particularly suitable for systems that must coordinate, adapt, and evolve alongside human partners (Kaelbling, Littman, and Moore 1996).

Recent research has applied RL to advance human–AI collaboration by enabling agents to align with human behaviour, generalise to new partners, and adapt to dynamic environments (Li et al. 2025). Core mechanisms supporting these goals include behaviour characterisation, that is, learning from human demonstrations or imitation as well as intention understanding, inferring human goals, motivations, or preferences, and multi-agent coordination, that is, learning to cooperate or communicate effectively in shared tasks. Such approaches allow AI agents to move beyond isolated optimisation toward behaviour that is contextually responsive to human collaborators. Comparing these approaches indicates trade-offs: behaviour-based methods offer adaptability, intention-based approaches enhance interpretability, and coordination mechanisms balance performance across diverse tasks (Li et al. 2025).

Empirical evidence supports the potential of RL in cooperative settings. Experiments combining RL with human-in-the-loop strategies show that systems incorporating human feedback through interventions or policy correction learn more efficiently and perform more effectively than those relying solely on demonstrations (Islam et al. 2025). When humans and RL agents collaborate, they tend to outperform humans or autonomous systems alone, achieving greater efficiency, diversity of strategies, and reduced cognitive workload. These findings demonstrate how human expertise and RL adaptability can complement each other to achieve superior outcomes.

The concept of complementarity provides a theoretical lens for understanding these results. Human–AI collaboration often succeeds when differences in information access or processing capabilities are leveraged rather than minimised (Hemmer et al. 2025). Humans typically contribute contextual reasoning, intuition, and ethical judgment, while AI systems contribute computational precision and scalability (Hemmer et al. 2025). Reinforcement learning supports this complementarity by enabling agents to learn policies that amplify human strengths or compensate for weaknesses, creating hybrid intelligence systems in which both sides continuously learn and improve through interaction (Dellermann et al. 2019). This mutual adaptation reflects the dynamic partnership envisioned in early symbiotic models (Taylor 1960).

At the same time, realising trustworthy and effective human–AI collaboration through RL raises important socio-technical challenges. Trust, transparency, and accountability are essential for ensuring that learning systems behave reliably and align with human values (Ramchurn, Stein, and Jennings 2021). RL agents capable of autonomous adaptation must remain interpretable and controllable, especially in safety-critical or high-stakes domains. Addressing these issues requires integrating insights from human–computer interaction, explainable AI, and responsible AI governance (Ramchurn, Stein, and Jennings 2021; Wang et al. 2020). Effective collaboration depends not only on algorithmic performance but also on clearly defined roles, shared goals, and adaptive autonomy that allows humans to retain meaningful oversight (Wang et al. 2020).

To systematically design and evaluate collaborative systems, recent frameworks have characterised human–AI interaction along three key dimensions: agency, interaction, and adaptation (Holter and El-Assady 2024). Reinforcement learning naturally spans these dimensions, sharing agency through negotiated control, enabling interaction via reward shaping and feedback, and supporting adaptation through continuous learning. Human-in-the-loop learning further strengthens this integration by embedding human feedback directly into the training process, improving efficiency, usability, and interpretability (Mosqueira-Rey et al. 2023). These developments establish RL as a central mechanism for enabling adaptive, human-centered AI systems that can learn from and with people.

2.2. Feedback mechanisms in reinforcement learning

Reinforcement learning relies fundamentally on feedback signals that enable an agent to learn optimal behaviour through trial-and-error interaction with its environment (Sutton and Barto 2018). In classical RL, this feedback is provided through a handcrafted reward function, a manually designed mapping from states and actions to scalar rewards. While effective for simple, well-defined problems, handcrafted rewards are increasingly recognised as insufficient for complex or open-ended tasks. Designing reward functions that capture nuanced objectives, social norms, or ethical principles is notoriously difficult and prone to failure. Even minor specification errors can lead to reward hacking, where the agent optimises for proxies that satisfy the numeric objective but contradict the intended goal (Christiano et al. 2017; Bignold et al. 2023; Toro Icarte et al. 2022).

This reward specification problem fundamentally limits the applicability of traditional RL. In high-dimensional or dynamic environments, such as autonomous driving, language modelling, or social robotics, defining a reward that consistently reflects desirable behaviour is infeasible. Handcrafted rewards are brittle, incomplete, and non-generalizable, often requiring extensive manual tuning. As a result, modern RL research increasingly seeks to learn reward functions from feedback, allowing the agent to infer the underlying objective rather than relying on pre-defined signals (Christiano et al. 2017; Bignold et al. 2023; Bıyık et al. 2022).

Recent developments have introduced several major paradigms for feedback-based learning, notably reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) (Bai et al. 2022), reinforcement learning from AI feedback (RLAIF) (Lee et al. 2024), and reinforcement learning with verifiable feedback (RLVF) (Trinh et al. 2024; Lightman et al. 2023). These paradigms differ fundamentally in the source, objectivity, and role of feedback in shaping agent behaviour. Table 1 summarises their respective feedback mechanisms, strengths, and limitations.

Reinforcement learning from human feedback emerged as a response to the reward specification problem. Instead of relying on manually crafted reward functions, RLHF leverages human judgments to train reward models that approximate user preferences. Typically, human evaluators compare pairs of agent-generated behaviours or responses and indicate which they prefer. These preference comparisons are then used to train a reward model, which serves as a learned objective for policy optimisation (Ouyang et al. 2022b). This paradigm has demonstrated that human intuition can effectively replace engineered reward signals, enabling agents to perform competitively in complex control and language tasks (Bıyık et al. 2022).

Table 1. Comparison of feedback mechanisms in reinforcement learning.

Paradigm	Feedback source	Strengths	Limitations
Reinforcement learning from human feedback (RLHF)	Human preference judgments	Captures subjective intent; flexible	Costly; inconsistent; optimises approval rather than correctness
Reinforcement learning from AI feedback (RLAIF)	AI-generated preferences	Scalable; multi-objective critics	Inherits model bias; lacks objective truth grounding
Reinforcement learning with verifiable feedback (RLVF)	Algorithmic verifiers	Objective; reproducible; correctness-grounded	Coverage-limited; risk of reward hacking; may reduce diversity

Reinforcement learning from human feedback inherits several major weaknesses. Human feedback is subjective, inconsistent, and expensive to collect (Bignold et al. 2023), with annotation quality varying widely across individuals. Linguistic preference judgments, while expressive, introduce ambiguity and context dependence that are difficult to formalise (Sumers et al. 2021). As a result, RLHF-trained models tend to align with perceived approval rather than objective correctness and remain dependent on the availability and consistency of human supervision (Xu et al. 2024).

To mitigate the scalability and cost limitations of RLHF, reinforcement learning from AI feedback replaces human evaluators with advanced language models or other AI systems that generate preference labels automatically (Lee et al. 2024). This substitution enables massive scaling of feedback collection while maintaining a similar training pipeline. Instead of human annotators ranking outcomes, AI models act as critics, scoring or ranking agent behaviours to produce training signals for the reward model. Empirical results show that AI-generated feedback can match or surpass human feedback in aligning LLMs, particularly in tasks emphasising helpfulness and harmlessness (Lee et al. 2024). Extensions of this paradigm introduce multi-objective variants and constitutional approaches, in which multiple AI evaluators assess different dimensions or guide self-critique through explicit ethical rules (Li et al. 2024; Williams 2024; Bai et al. 2022).

Despite these advances, RLAIF remains constrained by the subjectivity of its evaluators. AI critics inherit the biases, blind spots, and inconsistencies of their teacher models, and without objective grounding, preference-based feedback can propagate misinformation or overemphasise stylistic conformity over factual or logical validity. Empirical analyses further suggest that some gains attributed to RLAIF arise from capability gaps between teacher and student models rather than from reinforcement learning itself. In short, RLAIF scales preference-based training but does not resolve the epistemic limitations of subjective evaluation (Sharma et al. 2024).

Reinforcement learning with verifiable feedback offers a fundamentally different approach by grounding the reward signal in objective, verifiable truth rather than human or AI judgment. Instead of relying on preferences, RLVF employs automatically checkable feedback mechanisms derived from logical consistency, formal proofs, or deterministic validation criteria (Stojanovski et al. 2025; Wen et al. 2025). In this paradigm, the agent receives positive feedback only when its output satisfies verifiable correctness conditions, for example, when a mathematical proof passes formal verification, a programme executes successfully, or a reasoning chain yields a provably valid result. This framework eliminates

ambiguity and ensures that rewards correspond directly to truth-preserving outcomes rather than approximations of approval.

Large-scale environments based on RLVF principles demonstrate several advantages. Procedurally generated reasoning tasks with verifiable solutions enable unlimited, reproducible training data and remove the dependency on external annotators (Stojanovski et al. 2025). Empirical findings further indicate that RLVF-trained models exhibit more coherent and logically consistent internal reasoning, suggesting that verifiable rewards shape reasoning quality rather than merely surface behaviour (Wen et al. 2025).

By decoupling learning from subjective evaluators, RLVF directly addresses the instability and bias that plague RLHF and RLAIF. It provides a self-contained and truth-grounded feedback mechanism, enabling systems to improve autonomously while maintaining transparency and reproducibility.

2.3. Optimisation dynamics under verifiable feedback

Reward hacking occurs when an AI system learns to optimise its reward function in ways that increase measured performance while failing to achieve the designer's intended objective. In this sense, the system is not malfunctioning, it is successfully maximising the specified objective, thereby exposing a mismatch between what is measured and what is meant (Skalse et al. 2025). Although verifiable feedback addresses many limitations of preference-based training, reward hacking can arise when optimisation pressure concentrates behaviour on aspects of the reward that are explicitly encoded while neglecting dimensions that remain implicit (Pan, Bhatia, and Steinhardt 2022). Such effects intensify with optimisation strength: increasing model capacity or training duration amplifies even small mismatches between proxy and intended objectives. More generally, unless a reward function preserves the exact policy ordering of the true objective across the policy space, optimisation can concentrate probability mass on behaviours that score highly under the reward but underrepresent broader intent (Skalse et al. 2025).

In deterministic, rule-based settings, reward maximisation does not permit rule violation. However, the reward necessarily encodes only a subset of the intended objective space. Under Kullback–Leibler (KL)-regularized policy optimisation methods such as proximal policy optimization or group relative policy optimization, repeated updates amplify behaviours that reliably satisfy the encoded criteria. As a result, optimisation can induce systematic concentration on reward-secure solution patterns. This does not constitute classical proxy–true divergence; rather, it reflects over-optimisation of formally encoded properties.

Several structural dynamics follow from this mechanism:

- Concentration on high-reward templates that consistently satisfy all constraints.
- Preference for configurations that minimise violation risk under multiplicative penalties.
- Suppression of exploratory alternatives that increase reward variance.

These effects reflect the broader principle that optimisation amplifies whatever structure is present in the reward signal (Pan, Bhatia, and Steinhardt 2022). When the reward is deterministic and verifiable, maximal-reward outputs remain valid by construction. The residual vulnerability lies not in invalid behaviour but in narrowing of the output distribution when objective coverage is incomplete (Pan, Liang, and Lin 2026).

This phenomenon connects directly to recent analyses of mode collapse in reinforcement learning for large language models. In such settings, the optimal policy is mathematically determined by the reward, the reference distribution, and the regularisation strength. Under common hyperparameter regimes, the objective induces distributional sharpening: small reward differences produce exponential probability differences, and when multiple outputs receive equal reward, their relative weighting remains proportional to the reference model (GX-Chen et al. 2025). Reinforcement learning therefore reinforces dominant modes rather than encouraging coverage of all valid solutions.

Outcome-based RL systematically reduces diversity even when performance metrics improve (Song, Kempe, and Munos 2025). Reward-maximising objectives concentrate probability on high-reward reasoning paths rather than matching the full posterior over valid solutions (Hu et al. 2024). Token-level analyses show that reinforcement learning predominantly amplifies high-entropy decision points, reinforcing dominant trajectories while suppressing alternatives (S. Wang et al. 2025). Empirical studies consistently document reductions in lexical, semantic, and structural diversity as optimisation progresses (Slocum, Parker-Sartori, and Hadfield-Menell 2025). Under verifiable reinforcement, such convergence is not an adversarial loophole but a predictable consequence of optimisation over a partially specified objective. Mode collapse in this context is therefore better understood as distributional sharpening rather than reward hacking in the classical sense (Padmakumar and He 2024).

At the same time, several canonical forms of reward hacking are structurally precluded in this setting. First, amplification of reward-model estimation errors (Fluri et al. 2025) cannot arise because the reward is not learned but computed by a deterministic verifier. There is no statistical approximation whose errors could be exploited through optimisation. Second, evaluator manipulation, where a model increases perceived quality without improving task fidelity (Ngo, Chan, and Mindermann 2024; Liu et al. 2025), is eliminated because reward depends solely on externally defined, checkable criteria rather than subjective judgments. Third, spurious shortcut exploitation through incidental correlations in the environment (Skalse et al. 2025) is substantially constrained, since achieving maximal reward requires satisfying explicit validity conditions. Consequently, deterministic verifiable reinforcement shifts the remaining risk away from adversarial exploitation of the reward mechanism and toward incomplete coverage of the intended objective.

2.4. Functional decomposition as a case task

According to guideline VDI 2221 part 1 ("VDI 2019–1 Design of Technical Products and Systems – Model of Product Design" 2019), a *function* forms the central connection between a technical system's inputs, internal mechanisms, and outputs, thereby defining how the system fulfils its intended purpose. Expressing each function in an abstract, minimalist form (typically a concise verb–noun expression) helps designers avoid prematurely anchoring to specific solutions and instead encourages openness to alternative concepts.

When analyzing a technical product, its overall purpose is usually formulated as one or more *overall functions*. These are often visualised through a black-box model, which portrays the transformation of material, energy, and information flows from inputs to outputs. Such representations provide the foundation for functional modelling: the product is treated as an entity that accepts inputs, transforms them, and delivers outputs. A clear

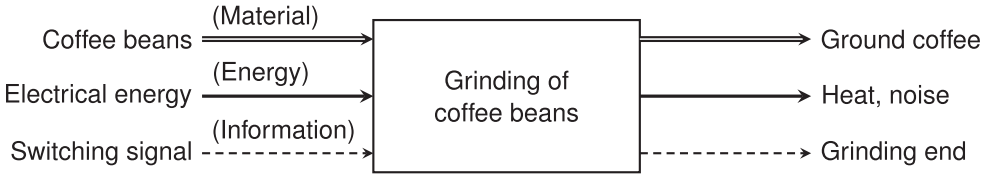

Figure 1. Example of the overall function for a coffee grinder (Schlattmann and Seibel 2021).

Table 2. Exemplary function list for a coffee grinder (Schlattmann and Seibel 2021).

Input functions (IF)	Internal functions (SF)	Output functions (OF)
IF 1 Receive coffee beans	SF 1 Feed coffee beans	OF 1 Dispense ground coffee
IF 2 Supply energy	SF 2 Grind coffee beans	OF 2 Release heat
IF 3 Switch on energy	SF 3 Store ground coffee	OF 3 Emit noise
…	…	…

definition of system boundaries together with explicit input–output specifications is widely recognised as a prerequisite for effective functional analysis (Ullman 2018). For instance, the basic operation of a coffee grinder can be shown in this manner, with raw beans and electrical power entering the system and ground coffee emerging as the output (Figure 1).

However, formulation of functions is not straightforward. Descriptions like 'grinding coffee beans' are too narrow, since countless variants of such tasks can exist. To address this, Roth (2000) introduced a systematic set of general functions. These functions cover material, energy, and information flows through operations such as *store, guide, convert,* and *transform*, complemented by *add* and *separate* to describe interactions across flow types. The result is a standardised catalogue of 30 functions that can be applied across domains. For example, the task 'actuate switch' can be reformulated more generally as 'add information to energy.' This taxonomy ensures consistency and repeatability in functional decomposition processes.

The overall function is typically broken down into smaller *subfunctions*, each representing a more manageable part of the task. There is no universal rule for how decomposition should proceed; the division depends on the design problem. A useful practice is to structure subfunctions in a table, distinguishing input, internal, and output operations. Decomposition continues until atomic functions are reached operations that can be tested independently and still maintain logical consistency with the larger system (Ullman 2018).

An exemplary function list for a coffee grinder is provided in Table 2.

To capture how subfunctions interact, they are arranged into functional structures, usually as block diagrams connected by flows. Material, energy, and information flows are distinguished by line type: double thin solid lines for material, bold solid lines for energy, and dashed lines for signals or feedback (Schlattmann and Seibel 2021). Such diagrams allow designers to explore alternative architectures by reordering or modifying subfunctions or by incorporating feedback. For a coffee grinder, for example, parallel and sequential relationships between subfunctions can be displayed in this way (Figure 2). Function structures are not expected to be unique or exhaustive; there may be multiple valid representations, but incorrect ones are possible.

The parallels between functional decomposition in engineering design and planning in artificial intelligence have also been noted (Rosenthal et al. 2024). In both cases, the

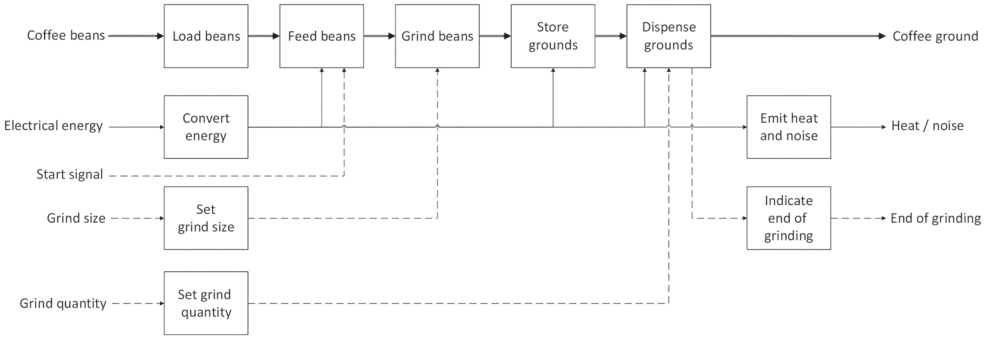

Figure 2. Possible function structure for a coffee grinder.

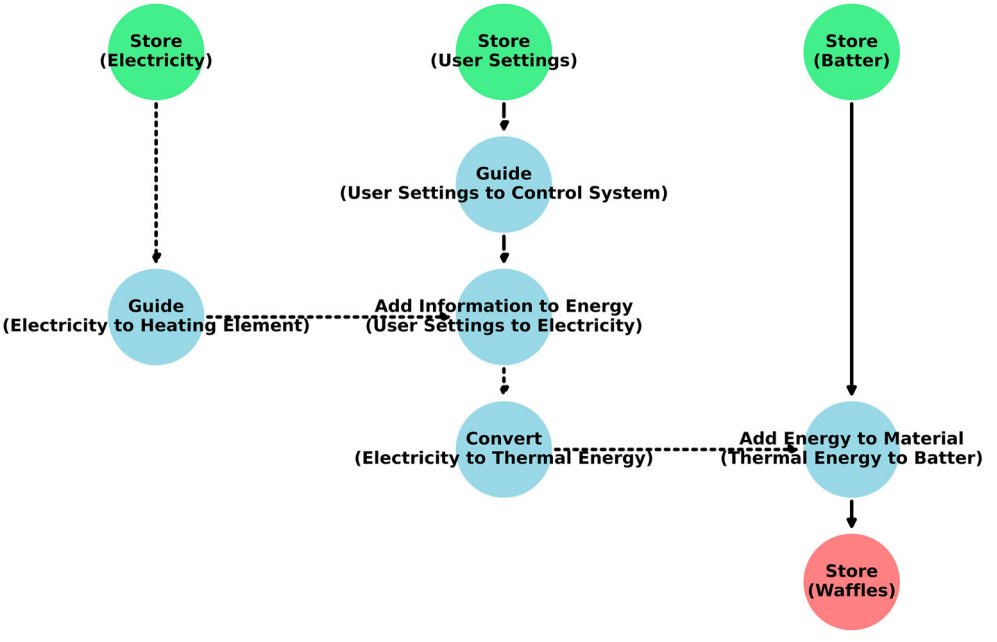

Figure 3. Functional decomposition of a waffle maker created by GPT-4o with Monte Carlo tree search (Haddad and Seibel 2025).

problem can be seen as describing state transitions: in engineering, functions alter system states, while in AI, actions alter world states. Interpreting FD as a planning problem allows the use of established AI planning techniques. By encoding the FD task in the planning domain definition language (Russell and Norvig 2020) and employing Roth's catalogue of functions (Roth 2000) as primitives, systematic function structures can be derived automatically (Rosenthal et al. 2024).

Recent experiments combine LLMs with symbolic planning methods. For instance, Haddad and Seibel (2025) integrate GPT-4o with a Monte Carlo tree search strategy to generate function structures from textual product descriptions with an example illustrated in Figure 3.

Table 3. Functional node types and structural constraints.

Function type	Inputs	Outputs	Max	Description
Store	1	1	1	Stores material, energy, or information
Guide	1	1	2	Directs a flow from source to target
Convert	1	1	2	Transforms one flow into another
Add	2	1	3	Merges two flows into one
Separate	1	2	3	Splits one flow into two outputs

While promising, results showed that more than half of the generated models still contained semantic or structural inconsistencies, underlining the difficulty of ensuring logical and physical validity in automated FD processes when relying solely on probabilistic models. This difficulty arises because functional decomposition is not an unconstrained generative task but a rule-governed modelling activity. Functional structures can be interpreted as directed, semantically typed graphs in which nodes represent functional operators (e.g. Store, Guide, Convert, Add, Separate) and edges represent flows of material, energy, or information. Each node type is associated with well-defined input–output constraints, as summarised in Table 3.

These type-specific constraints encode fundamental physical and logical principles of system behaviour, such as conservation, transformation, and branching of flows. As a result, the correctness of a functional decomposition is determined by explicit structural conditions –(e.g. admissible numbers of inputs and outputs and valid flow connectivity)– rather than by stylistic preference or subjective interpretation. This makes functional decomposition particularly well suited for rule-based verification and learning: domain expertise can be codified as deterministic constraints that reliably distinguish valid from invalid structures and can be integrated directly into adaptive learning processes.

3. Methodology

This study investigates reinforcement learning from verifiable feedback as a mechanism for adaptive, codified human–AI collaboration in generative intelligent design. Rather than treating reinforcement learning solely as a numerical optimisation procedure, we operationalise it as a structured interaction loop in which domain knowledge is encoded into deterministic verifiers that evaluate generated design artifacts against formal correctness rules. Learning is driven exclusively by verifiable feedback, without subjective human preference labels. The target task is functional decomposition, a foundational activity in engineering design that represents a technical system as a network of functions connected by material, energy, and information flows. Each FD is expressed as a directed graph composed of canonical operator types (e.g. Store, Guide, Convert, Add, Separate), subject to well-defined structural constraints on node connectivity and flow consistency.

Model outputs are represented in a structured JSON format containing (i) a set of functions with spatial metadata and (ii) three typed edge lists corresponding to material, energy, and information connections. This representation enables direct algorithmic verification of correctness properties such as connection validity, graph connectivity, and syntactic well-formedness. All experiments are conducted using Llama-3.1-8B-Instruct, loaded in 4-bit quantised form and adapted using QLoRA. The model is prompted with a fixed instruction template defining the FD grammar, operator semantics, and construction rules, followed

by four worked examples (coffee machine, Bluetooth speaker, toaster, television). Given a target product name, the model is instructed to generate a complete functional decomposition in JSON format and to return no extraneous text. Temperature sampling is used during training to encourage exploration, while deterministic decoding is used for evaluation to ensure reproducibility.

3.1. Verifiable feedback and reward formulation

A rule-based verifier encoding formal engineering design constraints inspired by functional modelling principles is used to evaluate each generated functional decomposition. Each FD is interpreted as a directed graph, where nodes correspond to functional elements labelled as Store, Guide, Convert, Add, or Separate, and directed edges represent energy, material, or information flows between functions. Verification operates on the structural properties of this graph, including node degree constraints and overall connectivity.

Prior to evaluation, all candidate outputs are normalised to a consistent schema to ensure robust and reproducible verification. This normalisation step enforces the presence of a function dictionary and three edge lists (energy, material, and information); missing or malformed fields are replaced with empty structures solely to enable safe rule evaluation, without modifying the semantic content of the generated decomposition. The pseudocode for the rule-based verifiable reward function is shown in Figure 4.

Verification proceeds by evaluating three classes of structural constraints. First, schema validity is enforced by requiring that the output be parseable as JSON and contain the required fields. Outputs violating this condition receive zero reward and are excluded from further evaluation. Second, operator arity constraints are evaluated by constructing a directed graph over all functions and computing node in-degrees and out-degrees. Valid arities are defined by engineering function semantics: Store operators must have exactly one incident connection, Guide and Convert operators must have exactly one incoming and one outgoing connection, Add operators must have two incoming and one outgoing connection, and Separate operators must have one incoming and two outgoing connections. Each operator violating its arity constraint contributes an additive penalty. Third, network integrity is evaluated by constructing an undirected version of the functional graph and computing its connected components. Disconnected subgraphs are penalised proportionally to their number, while fully connected networks receive a dedicated bonus.

The verifier maps these structural evaluations into a shaped scalar reward. A base reward is assigned for readable, well-formed JSON output, followed by size-based shaping terms proportional to the number of functional operators and connections. Penalties are applied for each operator-level violation and for each disconnected subgraph. Fully connected graphs receive an additional bonus, and decompositions that are both fully connected and free of operator-level errors receive a further correctness bonus. When structural violations are present, the total reward is multiplicatively damped to discourage partial compliance from dominating learning. The final reward is clipped to be non-negative.

3.2. Reinforcement learning via GRPO

Reinforcement learning is implemented using GRPO (Shao et al. 2024), interpreted here as RLVF. For each training step, the model generates a group of candidate FDs for a given

Algorithm 1 Deterministic reward computation for functional decomposition

Require: Candidate output c
Ensure: Scalar reward $r \geq 0$
$r \leftarrow 0$
JSON validity
if c is not parseable as JSON or does not follow the required schema **then**
return 0
end if
$r \leftarrow r + R_{\text{json}}$
Graph construction
$V \leftarrow$ set of functional operators in c
$E \leftarrow$ set of directed connections
Construct directed graph $D = (V, E)$
Construct undirected graph U by removing edge directions from D
Structural size reward
$r \leftarrow r + \alpha|V| + \beta|E|$
Operator arity constraints
$e \leftarrow 0$
for each operator $v \in V$ **do**
let $(d^-(v), d^+(v))$ be the in- and out-degree of v in D
if v is `Store` and $d^-(v) + d^+(v) \neq 1$ **then**
$e \leftarrow e + 1$
else if v is `Guide` or `Convert` and $(d^-(v), d^+(v)) \neq (1, 1)$ **then**
$e \leftarrow e + 1$
else if v is `Add` and $(d^-(v), d^+(v)) \neq (2, 1)$ **then**
$e \leftarrow e + 1$
else if v is `Separate` and $(d^-(v), d^+(v)) \neq (1, 2)$ **then**
$e \leftarrow e + 1$
end if
end for
$r \leftarrow r - \gamma e$
Network connectivity
$g \leftarrow \max(0, \text{number of connected components in } U - 1)$
$r \leftarrow r - \delta g$
Correctness bonuses
if $g = 0$ **then**
$r \leftarrow r + R_{\text{conn}}$
end if
if $e = 0$ and $g = 0$ **then**
$r \leftarrow r + R_{\text{correct}}$
end if
return $\max(0, r)$

Figure 4. Pseudocode of rule-based verifiable reward function.

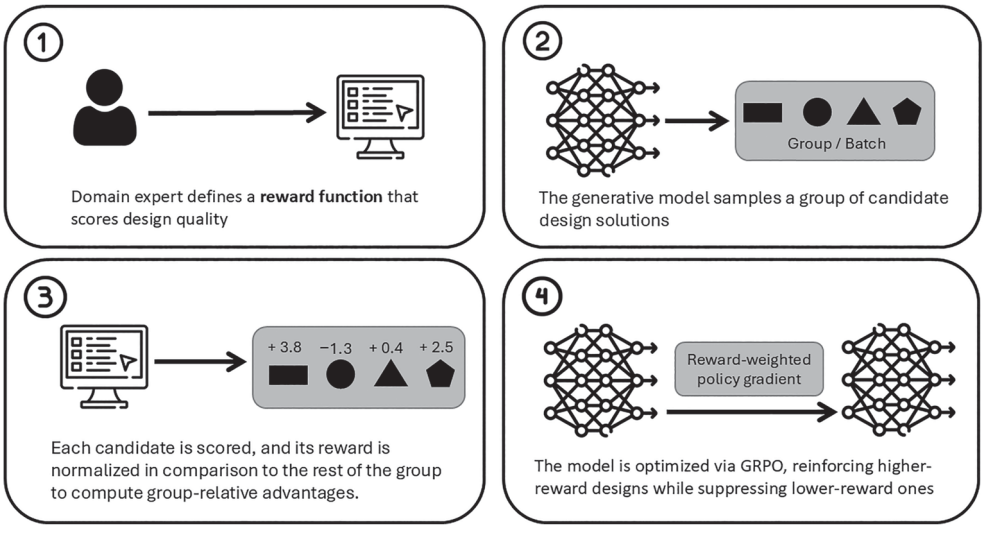

Figure 5. Codified collaboration framework through reinforcement learning.

product, each of which is evaluated independently by the rule-based verifier to produce a scalar reward. Advantages are computed relative to the mean reward within the group, and policy updates are performed using a log-probability-weighted objective. This group-based formulation stabilises learning by normalising rewards locally and eliminates the need for an explicit reward model.

Group relative policy optimisation is chosen over alternative policy optimisation methods for both conceptual and practical reasons. Proximal policy optimisation (Schulman et al. 2017) relies on value-function estimation and reward normalisation, which are unnecessary and potentially unstable in settings where rewards are sparse, discontinuous, and deterministically verifiable, as is the case for rule-based design correctness. Direct preference optimisation (Rafailov et al. 2024), while effective for static preference datasets, assumes a fixed set of pairwise comparisons and does not support iterative on-policy improvement when new verifiable feedback is generated during training. In contrast, GRPO directly leverages relative comparisons among multiple candidates sampled from the current policy, making it naturally compatible with verifier-driven learning loops. Adapter checkpoints and benchmark evaluations are saved periodically during training to enable longitudinal analysis of learning dynamics.

The visualisation of the procedure is illustrated in Figure 5.

3.3. Evaluation

For comparison, a supervised fine-tuned (SFT) baseline is trained on 257 verified functional decompositions using the same base model, LoRA configuration, and verifier logic. Training uses prompt masking so that loss is applied only to the decomposition output, not to the prompt or examples. Evaluation for SFT is performed on the same fixed 50-product test set used for GRPO, using deterministic decoding and the identical verifier. This ensures that differences between methods reflect learning paradigms rather than differences in evaluation or feedback.

Automated evaluation is conducted using the verifier on a fixed test set of 50 products, held constant across all models and training checkpoints. For each product, the model generates a single decomposition, which is scored for syntactic validity, structural correctness, and connectivity. Reported metrics include verifier reward statistics, percentage of correctly formatted outputs, percentage of fully connected decompositions, and percentage of error-free decompositions. All evaluations use the same verifier configuration to ensure comparability across GRPO, SFT, and baseline models.

To assess whether reinforcement learning induces structural homogenisation across products, we perform a population-level mode collapse analysis over training checkpoints. At each checkpoint, one decomposition per product is generated on the fixed 50-product test set. Each decomposition is assigned a coarse structural signature based on counts of operator types and flow-edge types between operators. Structural diversity is quantified using Shannon entropy over the signature distribution and a collapse ratio measuring the dominance of the most frequent structure. These metrics characterise how structural diversity evolves as verifier-aligned reward optimisation progresses.

To complement rule-based metrics, a qualitative evaluation is conducted on ten representative products. Functional decompositions are generated by four sources: the GRPO-trained model, the SFT model, GPT-4o using few-shot prompting, and graduate engineering students.

Twelve graduate-level participants with formal training in engineering design evaluated anonymized decompositions along three dimensions: completeness, logical coherence, and creativity, using a five-point Likert scale. The order of decompositions was randomised, and participants were blind to the source of each output. This evaluation captures aspects of design quality that are difficult to formalise algorithmically.

4. Results and discussion

Benchmark results from the functional decomposition experiments demonstrate that reinforcement learning from verifiable feedback substantially improved the model's structural validity and formatting consistency. The results are summarised in Table 4.

The base Llama-3.1-8B model produces the smallest functional decompositions, with an average of 17.2 nodes and 18.4 edges per system. GRPO training substantially increases structural richness: after 100 RL steps, the total number of nodes increases to 927, and further to 964 after 250 steps. The average number of nodes per system rises from 16.6 to 19.3,

Table 4. Benchmark results for functional decomposition experiments.

Metric	Llama-3.1-8B	Llama-3.1-8B 100th RL training	Llama-3.1-8B 250th RL training	Llama-3.1-8B-SFT	GPT-4o (Haddad & Seibel, 2025)
Total nodes	155	927	964	1023	1173
Total edges	166	940	916	1046	1084
Avg. nodes per system	17.2	16.6	19.3	20.9	23.5
Avg. edges per system	18.4	14.4	18.3	21.3	21.7
Avg. errors per system	2.0	1.3	0	1.2	2.1
Correct format	18%	92%	100%	98%	100%
Fully connected	10%	76%	100%	82%	44%
Error-free	4%	58%	100%	52%	16%

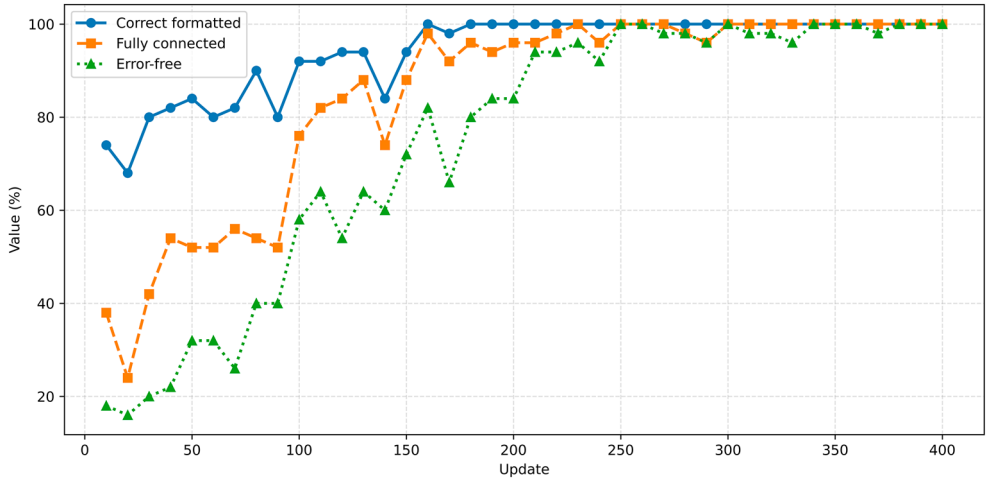

Figure 6. Performance evolution across reinforcement learning cycles of Llama-3.1-8B.

approaching the complexity of the SFT model (20.9) and GPT-4o (23.5). A similar trend is observed for edges.

The baseline model struggles with structural correctness, achieving only 18% correctly formatted outputs and 10% fully connected graphs. GRPO training leads to a sharp improvement: correct formatting rises to 92% at 100 steps and reaches 100% at 250 steps. Fully connected graphs increase to 76% at 100 steps and reach 100% at 250 steps. Average errors per system decrease consistently under GRPO training, dropping from 2.0 in the baseline to 1.3 at 100 steps and reaching zero at 250 steps. In contrast, the SFT model still exhibits residual structural errors (1.2 on average), while GPT-4o shows the highest error rate among the evaluated models (2.1). At 250 RL steps, the GRPO-trained model achieves 100% correct format, 100% fully connected outputs, and 100% error-free functional decompositions, outperforming both the SFT model and GPT-4o on all structural validity metrics. The metrics of the reinforcement learning cycles are displayed in Figure 6.

Despite its strong performance, the SFT model does not achieve full structural correctness. While its average complexity is comparable to the GRPO-trained model, it still produces disconnected graphs and residual rule violations. This highlights a fundamental limitation of purely supervised learning for structured generation: exposure to correct examples alone is insufficient to guarantee constraint satisfaction. Figure 8 illustrates a qualitative comparison between the baseline Llama-3.1-8B model (left) and the GRPO-trained model at 250 reinforcement updates (right) on the bread slicer task.

GPT-4o produces the largest and most complex decompositions but exhibits comparatively poor structural validity, including a low percentage of fully connected and error-free outputs. This suggests that high expressive capacity and fluency do not necessarily translate into structurally correct functional representations.

Figure 7 illustrates the evolution of the mean training and evaluation rewards over the course of GRPO-based RL training. In the early phase (updates 0–100), both curves exhibit a steep upward trend, indicating rapid policy improvement as the model learns to satisfy basic structural and formatting constraints.

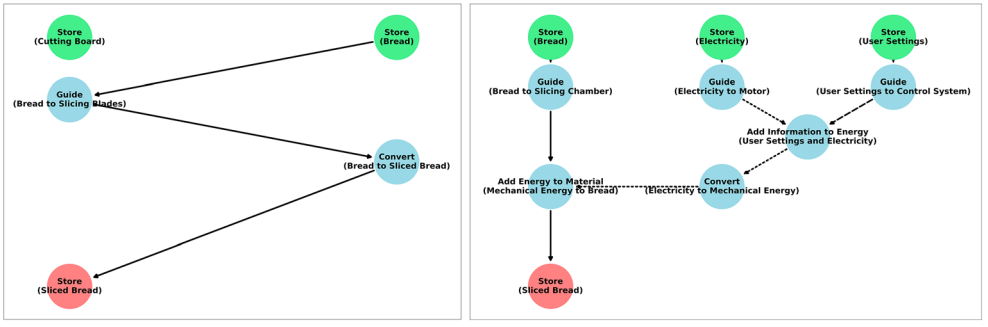

Figure 7. Comparison of bread slicer FDs: Llama 3.1 8B (left) vs Llama-3.1-8B 250 GRPO (right).

During this phase, the training reward shows high variance, reflecting exploratory behaviour and unstable policy updates, while the evaluation reward increases more smoothly. Between updates 150 and 250, the evaluation reward saturates and approaches a plateau near the maximum achievable reward, whereas the training reward continues to fluctuate with occasional sharp drops. This divergence suggests increasing overfitting to high-reward structures within the training distribution. Beyond update $\sim$250, both rewards stabilise at a high level, indicating convergence in terms of the reward function.

The computational complexity of the proposed approach differs substantially between supervised fine-tuning and reinforcement learning stages. During SFT, each optimiser update consists of a single forward and backward pass over one prompt–completion pair with prompt masking, resulting in an average time of 0.14 ± 0.02 s per update on a single GPU. In contrast, the RL stage performs a multi-sample optimisation cycle in which $K = 10$ candidate functional decompositions are generated per product prompt, followed by structured reward evaluation and a policy update. With gradient accumulation set to one, a single RL optimiser update therefore corresponds to a full sampling–evaluation–update cycle and requires 28.8 ± 0.2 s per update under the same hardware configuration.

4.1. Population-level diversity dynamics

While correctness metrics improve, diversity metrics reveal a contrasting trend. We track population-level diversity using three complementary indicators: structural entropy, the ratio of unique structures, and the collapse ratio (fraction of samples structurally identical to the batch mode).

Structural entropy decreases monotonically throughout training, with a measured slope of -0.0038 bits per update. Over the full training horizon, this corresponds to a reduction of more than 3.5 bits, implying a compression of the effective structural hypothesis space by approximately an order of magnitude. In parallel, the unique structure ratio exhibits a consistent downward trend (slope -0.00063 per update), indicating a progressive loss of structural variety across sampled decompositions. Conversely, the collapse ratio increases steadily (slope $+0.00062$ per update), showing that an increasing proportion of generated outputs converge toward a small number of repeated structural patterns.

Notably, the most pronounced entropy drop occurs at update 120, marking an early onset of population-level mode collapse. Beyond this point, diversity continues to decline even as reward and correctness continue to improve (Figure 9).

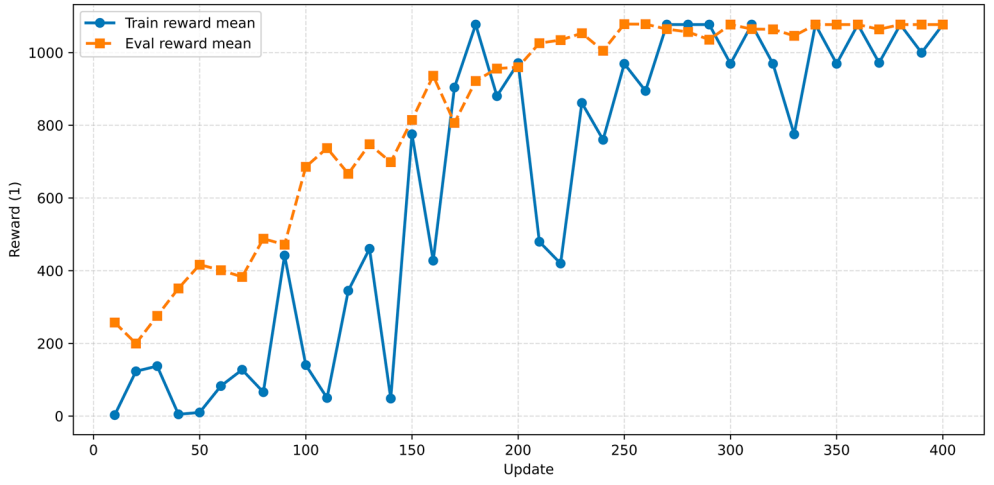

Figure 8. Train reward versus evaluation reward (GRPO).

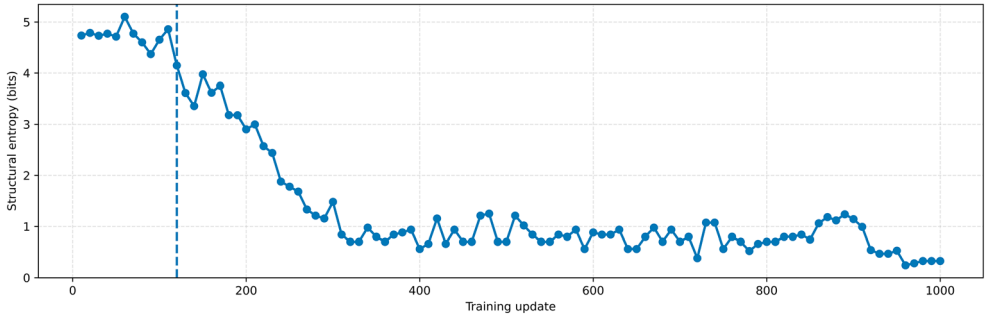

Figure 9. Population diversity over training.

We observe strong correlations between reward optimisation and diversity loss. Reward is highly negatively correlated with structural entropy (Pearson $r = -0.90$) and the unique structure ratio ($r = -0.94$), while showing a strong positive correlation with the collapse ratio ($r = +0.87$). These results indicate that improvements in reward are systematically associated with reductions in population-level diversity, but that the GRPO objective does not degenerate into trivial reward hacking.

Figure 10 shows a strong inverse relationship between reward and population-level diversity: as mean reward increases, the proportion of unique functional structures decreases sharply, confirming that mode collapse is a direct consequence of reward optimisation rather than a temporal artifact. Importantly, this collapse occurs despite stochastic sampling and temperature-based decoding, suggesting that it is not an artifact of inference determinism but rather a consequence of the learned policy distribution itself.

Our findings reveal a case of population-level mode collapse in reinforcement learning with verifiable rewards. Unlike classical mode collapse in generative modelling, which often manifests as repetitive surface-level outputs, the collapse observed here operates at the level of graph structure and functional organisation. Even when individual outputs remain valid and well-formed, the diversity of structural solutions collapses toward a small number

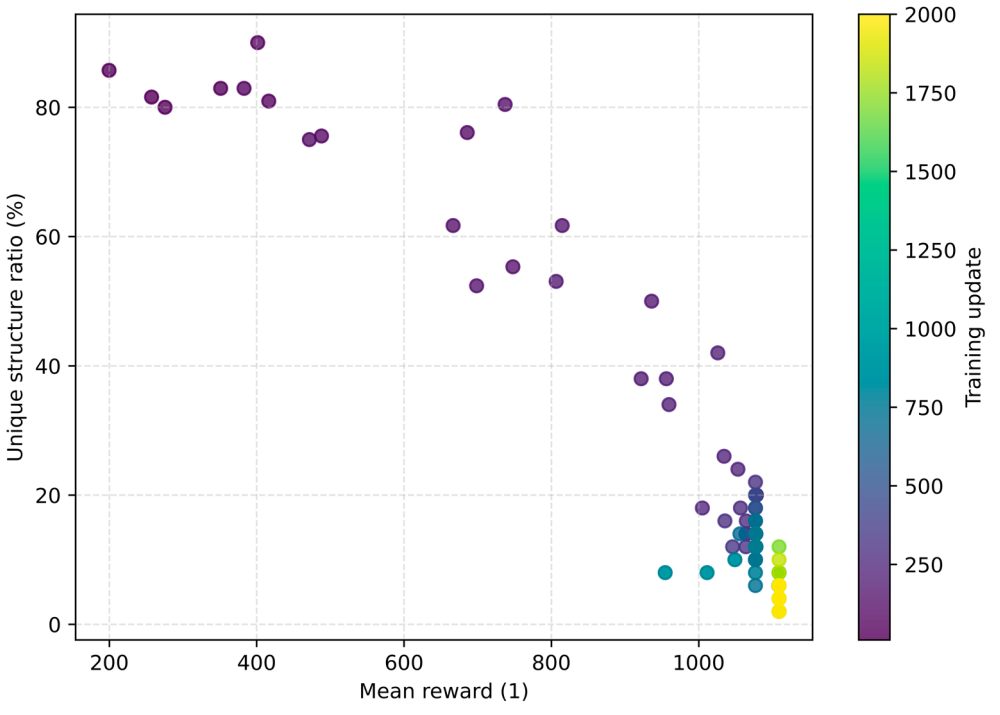

Figure 10. Reward – diversity tradeoff under GRPO.

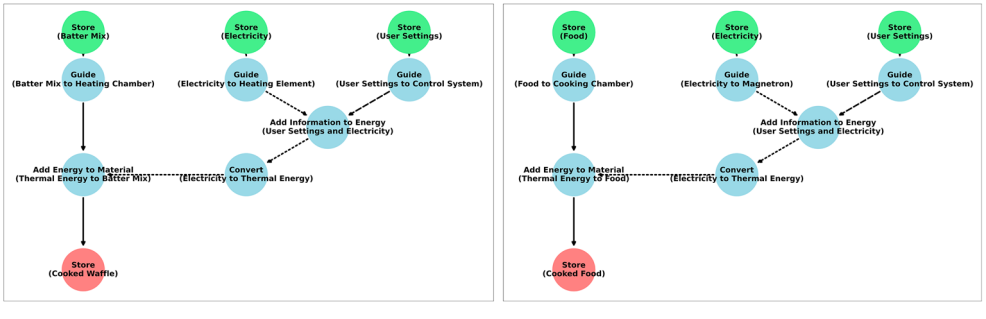

Figure 11. Comparison of waffle maker FD (left) vs electric convection oven (right).

of high-reward templates. This effect becomes visually apparent in Figure 11, which compares the functional decomposition of a waffle maker (left) and an electric convection oven (right). Although the products differ in internal mechanisms, the resulting decompositions exhibit similar structural layouts. A comparable structural pattern can also be observed in the right-hand example of Figure 7.

This phenomenon emerges early in training and persists throughout optimisation, indicating that it is not a transient instability but a stable attractor of the learning dynamics. The observed behaviour can be explained by the interaction between GRPO and deterministic reward functions. Verifiable rewards sharply distinguish correct from incorrect structures, creating a steep reward landscape in which a small subset of structural patterns dominates.

Once these patterns are discovered, gradient updates disproportionately reinforce them, suppressing alternative solutions even when they are semantically valid.

Crucially, standard GRPO objectives optimise expected reward without explicitly encouraging coverage of the solution space. As a result, the policy converges toward reward-maximising modes rather than maintaining a diverse population of solutions.

4.2. Qualitative performance

Human raters assessed each decomposition along three dimensions – completeness, logical correctness, and creativity – using a five-point Likert scale (1 = strongly disagree, 5 = strongly agree). This evaluation provides insight into whether the observed population-level mode collapse negatively affects perceived solution quality. Table 5 reports the mean ratings aggregated across all products and evaluators.

Across all three dimensions, both GRPO-trained models outperform the supervised fine-tuned baseline. For completeness, SFT achieves a mean score of 3.06, while GRPO with 100 updates improves slightly to 3.11, and GRPO with 250 updates reaches 3.13. Although these gains are modest, they indicate that reinforcement learning consistently reduces missing functional elements compared to purely supervised training. Nevertheless, both GRPO variants remain below GPT-4o (3.38) and the human baseline (3.40), suggesting that RL primarily refines existing structures rather than expanding functional coverage.

The strongest qualitative effect of GRPO is observed in logical correctness. SFT receives the lowest score in this dimension (3.01), indicating frequent issues in functional sequencing or causal structure. GRPO with 100 updates increases logical correctness to 3.16, and extended training to 250 updates yields a substantial further improvement to 3.32, the highest score among all automated systems. Notably, GRPO 250 even slightly exceeds the human baseline (3.27) in logical correctness, while GPT-4o achieves a comparable score (3.38). This result mirrors the quantitative findings, where GRPO achieves near-perfect rule compliance and connectivity.

In contrast, creativity shows a weaker and non-monotonic response to reinforcement learning. Supervised fine-tuning attains a mean creativity score of 2.54, which drops to 2.38 after 100 GRPO updates, indicating that early-stage RL training initially suppresses exploratory variation. With extended training, creativity recovers to 2.57 at 250 updates, marginally surpassing SFT but remaining below GPT-4o (2.67) and human participants (2.76). This pattern suggests that reinforcement learning favours safe and conventional functional decompositions rather than novel or unconventional ones.

When interpreted alongside the population-level collapse metrics, these qualitative results reveal an important nuance. Despite a pronounced reduction in structural entropy

Table 5. Performance comparison of GRPO-based reinforcement learning models against supervised fine-tuning, GPT-4o, and human baseline (mean ± SD).

Model	Completeness	Logic	Creativity
GRPO (100 updates)	3.11 ± 0.77	3.16 ± 0.80	2.38 ± 0.65
GRPO (250 updates)	3.13 ± 0.75	3.32 ± 0.77	2.57 ± 0.71
GPT-4o	3.38 ± 0.77	3.38 ± 0.75	2.67 ± 0.81
Supervised fine-tuning	3.06 ± 0.79	3.01 ± 0.78	2.54 ± 0.77
Student-generated	3.40 ± 0.76	3.27 ± 0.75	2.76 ± 0.84

and unique structure ratios during GRPO training, human raters continue to judge the generated decompositions as largely complete and logically correct. In particular, the increase in logical correctness from 3.01 (SFT) to 3.32 (GRPO 250) occurs concurrently with strong entropy decline, indicating that structural homogenisation does not negatively impact perceived correctness.

This apparent contradiction highlights the difference between population-level diversity and individual output quality. Mode collapse in this context does not imply that individual decompositions degrade in quality; rather, it reflects convergence toward a small set of highly reliable functional templates. From a human evaluation perspective, these templates are often sufficient to achieve high ratings in completeness and logical correctness.

Creativity, however, is more sensitive to this collapse. The consistently lower creativity scores for GRPO-trained models – 2.38–2.57 compared to 2.67 (GPT-4o) and 2.76 (humans) – provide qualitative confirmation that population-level homogenisation constrains perceived novelty and inventiveness. This aligns directly with the observed decreases in structural entropy and increases in collapse ratio.

The qualitative evaluation demonstrates that population-level mode collapse is not inherently detrimental when correctness and coherence are the primary objectives. GRPO successfully identifies a narrow set of decompositions that score highly under both formal structural rules and human judgments of logical soundness. Once these structures are discovered, further exploration offers limited benefit under the current reward formulation. This explains why GRPO 250 simultaneously achieves perfect structural correctness (100% error-free) and the high logical correctness score (3.32), despite exhibiting reduced diversity. In effect, reinforcement learning trades breadth of exploration for consistency and reliability, leading to high perceived performance even under conditions of structural convergence.

5. Conclusions and future work

This work positions reinforcement learning from verifiable feedback as a concrete and scalable mechanism for human–AI collaboration in generative engineering design. By encoding domain expertise into explicit, rule-based verifiers and integrating them directly into the learning loop, RLVF transforms human knowledge from static supervision into an adaptive and inspectable optimisation signal. This enables AI systems to improve autonomously while remaining aligned with formal engineering constraints.

Applied to the foundational task of functional decomposition, we demonstrate that verifiable reinforcement substantially improves structural validity over supervised fine-tuning, without requiring additional labelled data. Through GRPO-based optimisation, the model converges to fully connected, syntactically valid, and error-free functional representations, achieving performance levels that match or exceed strong baselines in logical correctness. Human evaluation confirms that these improvements are not merely formal but perceptually meaningful, particularly in terms of functional coherence and reasoning quality.

At the same time, our population-level analysis reveals a fundamental tradeoff inherent to verifiable reinforcement. While structural correctness improves monotonically, structural diversity decreases as training progresses, resulting in mode collapse at the level of functional graph structure. Importantly, this collapse does not correspond to a decline in

perceived quality. Instead, reinforcement learning identifies a small set of robust, reusable functional templates that reliably satisfy engineering constraints. From an engineering design perspective, this behaviour is often desirable: consistency, correctness, and predictability are frequently more valuable than unrestricted variability.

These results clarify why high performance can coexist with population-level collapse. In rule-governed design tasks, exploration beyond a certain point offers diminishing returns once valid solution schemas are discovered. RLVF thus naturally biases learning toward stable design patterns that encode expert knowledge in a reproducible form, effectively externalising human design heuristics into the model's policy.

Looking forward, a central direction for future work lies in extending verifiable reinforcement beyond purely local correctness objectives. First, reward functions that operate at the population level represent a critical extension. The current GRPO formulation optimises expected reward independently for each generated decomposition and does not account for redundancy across outputs. Future work should therefore incorporate group-aware or population-aware reward components (Anschel et al. 2025) that explicitly penalise structural similarity within a batch or across training iterations. Such rewards could operate on graph-level similarity metrics – such as node–edge overlap, structural Jaccard similarity (Alarcon 2019), or graph edit distance (Sanfeliu and Fu 1983) – and encourage coverage of multiple valid solution modes rather than convergence to a single dominant template.

Beyond population-level objectives, future work should also expand the semantic coverage of the verifier itself. While the current rule set enforces syntactic validity and structural connectivity, additional constraints could be formalised at the edge and operator level. For example, rules could ensure that functional flows remain type-consistent along edges, that transformations occur only at designated operator nodes, or that specific operator sequences are disallowed when they violate domain logic, such as redundant consecutive transformation steps. Such extensions would move verification from purely structural correctness toward structured semantic coherence. A further extension could introduce a secondary reward component that evaluates contextual plausibility of functional transformations. For instance, an auxiliary language-model-based critic could assess whether node-level transformations are physically meaningful within the given product context, while the primary deterministic verifier continues to enforce formal correctness. This hybrid formulation would preserve the transparency of rule-based validation while allowing gradual incorporation of higher-level domain semantics.

More broadly, this work advances a model of codified human–AI collaboration in generative engineering design. In our framework, domain expertise is not provided through subjective supervision but formalised directly as verifiable rules embedded in the reward function. Reinforcement learning then determines how these codified constraints are satisfied across diverse design instances. Our empirical results demonstrate that this paradigm yields structurally correct, logically coherent, and reproducible design representations, while revealing a principled tradeoff between correctness and diversity under KL-regularized optimisation.

The underlying principle is not limited to functional decomposition. Any generative engineering task in which feasibility is governed by deterministic validation procedures is structurally compatible with this framework. In parametric CAD design, geometric constraint systems already encode dimensional and relational consistency. In structured simulation-oriented design workflows, configuration models are validated through formal

rule-based checks prior to execution. In such domains, expert knowledge is already partially codified within validation mechanisms. Verifiable reinforcement transforms these validation systems into optimisation signals, allowing correctness criteria to directly guide learning. Wherever domain expertise can be formalised as explicit validation logic, it becomes directly optimisable through reinforcement learning.

In this sense, verifiable reinforcement represents more than a training technique; it defines an architectural principle for engineering AI systems. Experts specify what must be correct through formal, inspectable rules, and learning algorithms discover how those constraints can be satisfied efficiently and consistently. Future work in expanding rule coverage and incorporating population-aware objectives can further refine the balance between reliability and diversity. Together, these results position RLVF as a scalable foundation for correctness-driven generative design and structured, transparent human–AI collaboration in engineering practice.

Author contributions

CRediT: **Meno-Said Haddad:** Conceptualization, Data curation, Formal analysis, Methodology, Writing – original draft, Writing – review & editing; **Arthur Seibel:** Conceptualization, Resources, Writing – original draft, Writing – review & editing

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data that support the findings of this study are openly available in GitHub at https://github.com/IPTS-PRODUCT-DESIGN/CC-RLVF

Generative AI statement

ChatGPT (version 5, OpenAI) was used solely for linguistic polishing and for sharpening ideas that had already been developed; all substantive decisions remained with the authors.

ORCID

Meno-Said Haddad https://orcid.org/0009-0000-2080-088X
Arthur Seibel https://orcid.org/0000-0003-3989-9626

References

Alarcon, N. 2019. "*Similarity in Graphs: Jaccard versus the Overlap Coefficient*." https://Developer.Nvidia.Com/Blog/Similarity-in-graphs-Jaccard-versus-the-Overlap-Coefficient.

Anschel, O., A. Shoshan, A. Botach, S. H. Hakimi, A. Gendler, E. Ben Baruch, N. Bhonker, I. Kviatkovsky, M. Aggarwal, and G. Medioni. 2025. "Group-Aware Reinforcement Learning for Output Diversity in Large Language Models." Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 32382–32403).

Bai, Y., A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, et al. 2022. *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback*. http://arxiv.org/abs/2204.05862.

Bai, Y., S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, et al. 2022. *Constitutional AI: Harmlessness from AI Feedback*. http://arxiv.org/abs/2212.08073.

Bignold, A., F. Cruz, M. E. Taylor, T. Brys, R. Dazeley, P. Vamplew, and C. Foale. 2023. "A Conceptual Framework for Externally-Influenced Agents: An Assisted Reinforcement Learning Review." *Journal of Ambient Intelligence and Humanized Computing* 14 (4): 3621–3644. https://doi.org/10.1007/s12652-021-03489-y.

Bıyık, E., D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. 2022. "Learning Reward Functions from Diverse Sources of Human Feedback: Optimally Integrating Demonstrations and Preferences." *The International Journal of Robotics Research* 41 (1): 45–67. https://doi.org/10.1177/02783649211041652.

Bordas, A., P. Le Masson, M. Thomas, and B. Weil. 2024. "What Is Generative in Generative Artificial Intelligence? A Design-Based Perspective." *Research in Engineering Design* 35 (4): 427–443. https://doi.org/10.1007/s00163-024-00441-x.

Bougie, N., T. Onishi, and Y. Tsuruoka. 2024. "Interpretable Imitation Learning with Symbolic Rewards." *ACM Transactions on Intelligent Systems and Technology* 15 (1): 1–34. https://doi.org/10.1145/3627822.

Christiano, P. F., J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. 2017. "Deep Reinforcement Learning from Human Preferences." Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 4302–4310).

Dellermann, D., P. Ebel, M. Söllner, and J. M. Leimeister. 2019. "Hybrid Intelligence." *Business & Information Systems Engineering* 61 (5): 637–643. https://doi.org/10.1007/s12599-019-00595-2.

Feuerriegel, S., J. Hartmann, C. Janiesch, and P. Zschech. 2024. "Generative AI." *Business & Information Systems Engineering* 66 (1): 111–126. https://doi.org/10.1007/s12599-023-00834-7.

Fluri, L., L. Lang, A. Abate, P. Forré, D. Krueger, and J. Skalse. 2025. *The Perils of Optimizing Learned Reward Functions: Low Training Error Does Not Guarantee Low Regret*. https://doi.org/10.48550/arXiv.2406.15753

GX-Chen, A., J. Prakash, J. Guo, R. Fergus, and R. Ranganath. 2025. *KL-Regularized Reinforcement Learning Is Designed to Mode Collapse*. https://doi.org/10.48550/arXiv.2510.20817

Haddad, M.-S., and A. Seibel. 2025. "Functional Decomposition of Technical Products Based on Large Language Models and Monte Carlo Tree Search." *Proceedings of the Design Society* 5:1913–1922. https://doi.org/10.1017/pds.2025.10205.

Hemmer, P., M. Schemmer, N. Kühl, M. Vössing, and G. Satzger. 2025. "Complementarity in Human-AI Collaboration: Concept, Sources, and Evidence." *European Journal of Information Systems* 34 (6): 979–1002. https://doi.org/10.1080/0960085X.2025.2475962.

Holter, S., and M. El-Assady. 2024. "Deconstructing Human-AI Collaboration: Agency, Interaction, and Adaptation." *Computer Graphics Forum* 43 (3): e15107. https://doi.org/10.1111/cgf.15107.

Hu, E. J., M. Jain, E. Elmoznino, Y. Kaddar, G. Lajoie, Y. Bengio, and N. Malkin. 2024. *Amortizing Intractable Inference in Large Language Models*. https://doi.org/10.48550/arXiv.2310.04363.

Islam, M. S., S. Das, S. K. Gottipati, W. Duguay, C. Mars, J. Arabneydi, A. Fagette, M. Guzdial, and M. E. Taylor. 2025. "Human-AI Collaboration in Real-World Complex Environment with Reinforcement Learning." *Neural Computing and Applications* 37 (23): 18957–18987. https://doi.org/10.1007/s00521-025-11288-1.

Kaelbling, L. P., M. L. Littman, and A. W. Moore. 1996. "Reinforcement Learning: A Survey." *Journal of Artificial Intelligence Research* 4 (1): 237–285.

Lee, H., S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. Lu, C. Bishop, et al. 2024. "RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback." RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback (pp. 26874–26901).

Li, A., Q. Xiao, P. Cao, J. Tang, Y. Yuan, Z. Zhao, X. Chen, et al. 2024. *HRLAIF: Improvements in Helpfulness and Harmlessness in Open-domain Reinforcement Learning From AI Feedback*. http://arxiv.org/abs/2403.08309.

Li, W., H. Liu, K. Huang, and A. Hussain. 2025. "Reinforcement Learning for Human-AI Collaboration: Challenges, Mechanisms, and Methods." *Cognitive Computation* 17 (5): 146. https://doi.org/10.1007/s12559-025-10500-7.

Lightman, H., V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. 2023. *Let's Verify Step by Step*. http://arxiv.org/abs/2305.20050.

Liu, Q., H. Xu, X. Chen, W. Chen, Y. W. Teh, and N. Miao. 2025. *Enhancing Large Language Model Reasoning with Reward Models: An Analytical Survey*. https://doi.org/10.48550/arXiv.2510.01925

Mosqueira-Rey, E., E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á Fernández-Leal. 2023. "Human-in-the-loop Machine Learning: A State of the art." *Artificial Intelligence Review* 56 (4): 3005–3054. https://doi.org/10.1007/s10462-022-10246-w.

Mroueh, Y. 2025. *Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification*. http://arxiv.org/abs/2503.06639.

Ngo, R., L. Chan, and S. Mindermann. 2024. "The Alignment Problem from a Deep Learning Perspective." The Twelfth International Conference on Learning Representations.

Ouyang, L., J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, et al. 2022. "Training Language Models to Follow Instructions with Human Feedback." *Advances in Neural Information Processing Systems* 35:27730–27744.

Padmakumar, V., and H. He. 2024. "Does Writing with Language Models Reduce Content Diversity?." The Twelfth International Conference on Learning Representations.

Pan, A., K. Bhatia, and J. Steinhardt. 2022. *The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models*. https://doi.org/10.48550/arXiv.2201.03544

Pan, P.-C., Y. Liang, and S. Lin. 2026. *Reward Modeling for Reinforcement Learning-Based LLM Reasoning: Design, Challenges, and Evaluation*. https://doi.org/10.48550/arXiv.2602.09305

Rafailov, R., A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. 2024. "Direct Preference Optimization: Your Language Model Is Secretly a Reward Model." *Advances in Neural Information Processing Systems* 36: 53728–53741.

Ramchurn, S. D., S. Stein, and N. R. Jennings. 2021. "Trustworthy Human-AI Partnerships." *IScience* 24 (8): 102891. https://doi.org/10.1016/j.isci.2021.102891.

Regenwetter, L., A. H. Nobari, and F. Ahmed. 2022. "Deep Generative Models in Engineering Design: A Review." *Journal of Mechanical Design* 144 (7): 071704. https://doi.org/10.1115/1.4053859.

Rosenthal, P., N. Demke, F. Mantwill, and O. Niggemann. 2024. "Plan-Based Derivation of General Functional Structures in Product Design." IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)https://doi.org/10.1109/ICPS59941.2024.10640039.

Roth, K. 2000. *Konstruieren mit Konstruktionskatalogen*. Berlin: Springer. https://doi.org/10.1007/978-3-642-17466-7.

Russell, S., and P. Norvig. 2020. *Artificial Intelligence: A Modern Approach 3rd Edition*. Upper Saddle River: Pearson.

Saadi, J. I., and M. C. Yang. 2023. "Generative Design: Reframing the Role of the Designer in Early-Stage Design Process." *Journal of Mechanical Design* 145 (4): 041411. https://doi.org/10.1115/1.4056799.

Sanfeliu, A., and K.-S. Fu. 1983. "A Distance Measure between Attributed Relational Graphs for Pattern Recognition." *IEEE Transactions on Systems, Man, and Cybernetics, SMC* 13 (3): 353–362. https://doi.org/10.1109/TSMC.1983.6313167.

Schlattmann, J., and A. Seibel. 2021. *Structure and Organization of Product Development Projects*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-81046-7.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. *Proximal Policy Optimization Algorithms*. https://doi.org/10.48550/arXiv.1707.06347

Shao, Z., P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, et al. 2024. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. https://doi.org/10.48550/arXiv.2402.03300

Sharma, A., S. Keh, E. Mitchell, C. Finn, K. Arora, and T. Kollar. 2024. "A Critical Evaluation of AI Feedback for Aligning Large Language Models." Proceedings of the 38th International Conference on Neural Information Processing Systems (pp. 29166–29190).

Skalse, J., N. H. R. Howe, D. Krasheninnikov, and D. Krueger. 2025. "Defining and Characterizing Reward Hacking." *Advances in Neural Information Processing Systems* 35: 9460–9471.

Slocum, S., A. Parker-Sartori, and D. Hadfield-Menell. 2025. "Diverse Preference Learning for Capabilities and Alignment." The Thirteenth International Conference on Learning Representations.

Song, Y., J. Kempe, and R. Munos. 2025. *Outcome-based Exploration for LLM Reasoning*. https://doi.org/10.48550/arXiv.2509.06941

Stojanovski, Z., O. Stanley, J. Sharratt, R. Jones, A. Adefioye, J. Kaddour, and A. Köpf. 2025. *REASONING GYM: Reasoning Environments for Reinforcement Learning with Verifiable Rewards*. http://arxiv.org/abs/2505.24760.

Sumers, T. R., M. K. Ho, R. D. Hawkins, K. Narasimhan, and T. L. Griffiths. 2021. "Learning Rewards from Linguistic Feedback." Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 7, pp. 6002–6010).

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge: A Bradford Book.

Taylor, R. W. 1960. "Man-Computer Symbiosis." *Ire Transactions on Human Factors in Electronics* 1:4–11.

Toro Icarte, R., T. Q. Klassen, R. Valenzano, and S. A. McIlraith. 2022. "Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning." *Journal of Artificial Intelligence Research* 73:173–208. https://doi.org/10.1613/jair.1.12440.

Trinh, T. H., Y. Wu, Q. V. Le, H. He, and T. Luong. 2024. "Solving Olympiad Geometry without Human Demonstrations." *Nature* 625 (7995): 476–482. https://doi.org/10.1038/s41586-023-06747-5.

Ullman, D. G. 2018. *The Mechanical Design Process*. New York: McGraw-Hill.

VDI 2221-1 Design of technical products and systems – Model of product design. 2019. Düsseldorf: VDI

Verganti, R., L. Vendraminelli, and M. Iansiti. 2020. "Innovation and Design in the Age of Artificial Intelligence." *Journal of Product Innovation Management* 37 (3): 212–227. https://doi.org/10.1111/jpim.12523.

Wang, D., E. Churchill, P. Maes, X. Fan, B. Shneiderman, Y. Shi, and Q. Wang. 2020. "From Human-Human Collaboration to Human-AI Collaboration." Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systemshttps://doi.org/10.1145/3334480.3381069.

Wang, S., L. Yu, C. Gao, C. Zheng, S. Liu, R. Lu, K. Dang, et al. 2025. "Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning." The Thirty-ninth Annual Conference on Neural Information Processing Systems.

Wen, X., Z. Liu, S. Zheng, S. Ye, Z. Wu, Y. Wang, Z. Xu, et al. 2025. *Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Reasoning in Base LLMs*. http://arxiv.org/abs/2506.14245.

Williams, M. 2024. *Multi-objective Reinforcement learning from AI Feedback*. http://arxiv.org/abs/2406.07295.

Xu, W., B. An, S. Yan, and Z. Xu. 2024. "Reinforcement Learning from Diverse Human Preferences." Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (pp. 5262–5270).

Zhu, Q., and J. Luo. 2022. "Generative Transformers for Design Concept Generation." *Journal of Computing and Information Science in Engineering* 23 (4): 041003. https://doi.org/10.1115/1.4056220.