

Article

Bridging Cognitive and Expression Spaces in Creative AI by Integrating DIKWP-TRIZ and Semantic Mathematics

Zhendong Guo ¹ and Yucong Duan ^{2,*}

¹ School of Information and Communication Engineering, Hainan University, Haikou 570228, China; guozhendong@hainanu.edu.cn

² School of Computer Science and Technology, Hainan University, Haikou 570228, China

* Correspondence: duanyucong@hainanu.edu.cn

Abstract

Large Language Models (LLMs) generate fluent text but often struggle with reliable multi-step reasoning, factual grounding, and stable use of long context, especially when inputs are incomplete, inconsistent, or imprecise. To address these challenges, we propose a Creative AI framework that integrates DIKWP-TRIZ with a semantic-mathematical constraint layer. DIKWP-TRIZ extends TRIZ by embedding a DIKWP (Data–Information–Knowledge–Wisdom–Purpose) network, enabling purposeful, value-aware transformations and explicit repair operations under 3-No conditions. The semantic layer introduces three context-indexed constraints over concept–expression mappings (Existence, Contextual Uniqueness, and Transitivity), making ambiguities and contradictions explicit and checkable during inference and generation. We enumerate the $\text{DIKWP} \times \text{DIKWP}$ transformation type space (25 ordered pairs over $\{D, I, K, W, P\}$) and provide candidate TRIZ inventive principles for each type as design-time guidance. A global Purpose controller steers transformation selection and enforces goal alignment and ethical constraints. We present a reference architecture and qualitative case analyses against a standard LLM, illustrating how the framework structures intermediate steps, surfaces assumptions, and supports traceable explanations. Quantitative benchmarking remains for future work.

Keywords: DIKWP-TRIZ; semantic reasoning; semantic mathematics; LLMs

1. Introduction

LLMs have achieved remarkable success in generating coherent text. However, they still struggle with deeper cognitive understanding and consistency. Current LLMs operate primarily by predicting the next token in a high-dimensional language embedding space. This mechanism enables fluent and coherent output. However, such fluency does not necessarily imply, nor does it guarantee, true comprehension [1–4]. As a result, LLMs can produce factual inconsistencies or hallucinations [5–7], have difficulty maintaining long-term context [8], and often fail at common sense reasoning [9,10]. In contrast, human cognition spans a rich cognitive space of perceptions, concepts, and goals, enabling purposeful and creative problem-solving beyond mere word prediction [1]. Bridging the gap between the human cognitive space and the LLM expression space is therefore essential for developing Creative AI systems capable of human-like reasoning and innovation [11–13].

DIKWP-TRIZ has emerged as a promising approach to address this gap. TRIZ is a well-established systematic methodology for innovation and inventive problem-solving [14,15]. DIKWP-TRIZ extends the classical TRIZ by incorporating the DIKWP model into the



Academic Editor: Valeri Mladenov

Received: 12 January 2026

Revised: 23 February 2026

Accepted: 24 February 2026

Published: 26 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

innovation process [16]. By integrating human-centric elements like Purpose and Wisdom into TRIZ, the DIKWP-TRIZ framework emphasizes value-driven, cognitive problem solving and is better equipped to tackle problems characterized by the “3-No”. In parallel, semantic mathematics has been proposed as a formalism to embed semantic meaning into mathematical structures. This approach links cognitive, semantic, and Expression Spaces under rigorous axiomatic constraints, aiming to reduce ambiguity and ensure logically consistent transformations [17]. By introducing formal semantic axioms that govern how concepts in the cognitive space map onto representations in the LLM expression space, semantic mathematics offers a principled framework for ensuring consistency and completeness in semantic transformations.

In this paper, we unify DIKWP-TRIZ and semantic mathematics into a single framework. The framework models reasoning and generation as sequences of DIKWP-typed transformations guided by TRIZ patterns, while a semantic verifier enforces checkable, context-indexed constraints on concept–expression mappings. This design aims to reduce uncontrolled assumption propagation and to improve traceability under 3-No problems.

The main contributions of this work are as follows:

- **Integration of DIKWP-TRIZ with Semantic Mathematics:** We formulate a unified framework where TRIZ’s inventive problem-solving principles are applied within and between DIKWP elements, under formal semantic constraints. This bridges cognitive modeling and expression generation, enabling AI to handle complex tasks with both creativity and semantic rigor.
- **Modeling 3-No Problems in Input/Output:** We formally characterize how incompleteness, inconsistency, and imprecision manifest in LLM inputs and outputs, and extend the DIKWP-TRIZ model to address each issue. By mapping the 3-No problems onto DIKWP’s Cognitive, Semantic, and Expression spaces, the framework offers robust strategies for resolving ambiguity and contradictions in queries and responses.
- **DIKWP $\times$ DIKWP Cognitive Transformations:** We define 25 possible transformations between each pair of DIKWP elements, each guided by TRIZ principles. This complete enumeration over $L \times L$ provides a structural index of DIKWP-typed transformation categories; for each category, we list candidate TRIZ principles as design-time guidance.
- **Semantic Axioms for Consistency and Completeness:** We introduce formal axioms within the semantic mathematics framework. We derive how these axioms enforce expression-space consistency and completeness in the LLM’s output.
- **Purpose-Driven Inference and Ethical Alignment:** By including the “Purpose” dimension from DIKWP, the framework is designed to steer AI reasoning with goal-oriented and ethical considerations. We propose an architecture where the Purpose component evaluates and guides inference steps, ensuring the AI’s solutions are not only technically sound but also aligned with intended goals and human values.

The remainder of this paper is organized as follows. Section 2 reviews related work on TRIZ, cognitive modeling in AI, and DIKWP. Section 3 presents the proposed theoretical framework, including the DIKWP $\times$ DIKWP transformation mechanisms, and its integration with semantic mathematics. Section 4 formalizes the 3-No problems in LLM inputs and outputs. Section 5 introduces the semantic axioms and corresponding transformation rules that ensure consistency and completeness. Section 6 describes the overall architecture and provides an implementation blueprint, along with an illustrative example and an evaluation protocol for future benchmarking. Section 7 discusses the framework’s capabilities, implications, and limitations. Finally, Section 8 concludes the paper and outlines future research directions.

2. Related Work

2.1. TRIZ and Its Limitations in Cognitive Problems

TRIZ, developed by Altshuller and colleagues, is a methodology originally devised for technological innovation. It provides systematic tools, such as 40 inventive principles (see Appendix A Table A1), and contradiction matrices to resolve conflicts in engineering design. TRIZ has been successfully applied in engineering, manufacturing, and even business process innovation [14,18]. However, traditional TRIZ is grounded in technical contexts and assumes well-defined problems; it requires expert knowledge to apply and often does not translate easily to domains like software or cognitive tasks [19–21]. A key limitation is that classical TRIZ does not explicitly account for incomplete or inconsistent information in problem definitions, nor does it incorporate human value judgments during solution generation. Ilevbare et al. [19] identified that TRIZ's complexity and need for extensive knowledge hinder its broader effectiveness, and that dealing with incomplete, inconsistent, or imprecise content remains challenging for standard TRIZ approaches. For example, applying TRIZ in multi-domain contexts can lead to semantic mismatches and difficulties in knowledge base management when information is noisy or partially missing. These limitations motivate extending TRIZ with frameworks that handle uncertainty and cognition.

Recent efforts to modernize TRIZ include ontology-based inventive problem solving and computer-aided TRIZ systems [22–25]. Ontology-driven approaches link diverse knowledge sources via semantic similarity to resolve information inconsistencies [26,27]. While promising, such methods depend on precise ontologies and struggle with scale and dynamically changing data [16]. For instance, Yan et al. developed an ontology-based TRIZ method and an automated system, “Ingenious TRIZ”, to assist inventive problem solving [22,27]. These tools improve knowledge integration but rely on complex semantic computations and cannot cover all scenarios without manual intervention [16]. This underscores the need for a more flexible, cognition-oriented extension of TRIZ that can handle the “3-No” problems inherent in real-world, especially AI-related, problems.

2.2. Cognitive Modeling in AI and DIKWP Advancements

Bridging human cognitive processes with AI systems has been a long-standing goal in AI research. ACT-R and Soar are among the most established cognitive architectures, providing explicit mechanisms for declarative/procedural memory, goal representation, production-rule execution, and learning, and they have been widely used as testable accounts of cognition and problem solving [28–31]. Comparative surveys of cognitive architectures place these systems within a broader landscape and highlight recurring design trade-offs between symbolic transparency, engineering effort, and breadth of task coverage [32,33]. These precedents motivate our emphasis on explicit intermediate DIKWP artifacts and traceability, while also informing our choice to leverage LLMs for scalable expression generation.

Recent work on LLM-based agents has begun to incorporate memory and planning components, but these additions often remain weakly grounded and hard to verify formally [34–38]. However, large data-driven models like contemporary LLMs lack an explicit cognitive model; they operate on statistical correlations in language rather than grounded, experiential knowledge [39,40]. As a result, LLMs may generate plausible-sounding text that nonetheless lacks true understanding or can be misaligned with real-world context [41,42]. Researchers have noted that human cognition involves multimodal perception and an evolving semantic understanding that current language models do not fully replicate [43,44]. This gap has led to explorations of Expression Spaces and semantic

frameworks that connect the sub-symbolic learning of neural networks with higher-order symbolic reasoning and meaning representation [44–48].

In parallel, neuro-symbolic AI has, for decades, pursued the integration of sub-symbolic learning with symbolic rules and logic-based inference—ranging from early neural-symbolic cognitive reasoning frameworks to more recent surveys connecting graph neural networks, differentiable reasoning, and logic constraints [49–52]. The previous literature is closely related to our goals: our semantic mathematics layer treats concept-expression mappings and DIKWP transformations as checkable constraints rather than purely implicit language model behavior, thereby providing a verifiable interface between internal semantic representations and LLM-based outputs.

The DIKW model (Data–Information–Knowledge–Wisdom) has historically been used as a hierarchy in knowledge management, describing increasing levels of understanding [53,54]. Recent work by Duan et al. extends this to DIKWP by adding Purpose as the pinnacle element, forming a purpose-driven cognitive model [17,55]. The inclusion of Purpose explicitly brings in the role of intentions, goals, and value judgments in processing information. The DIKWP framework provides a structured conceptualization of cognition: Data corresponds to raw facts or sensory inputs, Information to structured or contextualized data, Knowledge to applied information (rules, models), Wisdom to higher-order insight (e.g., ethical or social understanding), and Purpose to the underlying goals or motivations driving decisions [17,56–60]. This extended DIKWP model has been applied in domains such as artificial consciousness, data privacy, and human–AI interactions, demonstrating its versatility in handling complex information processing across domains [61–64]. Notably, refs. [57–60] were reported in the same conference and thematic session of HPCC/DSS/SmartCity/DependSys, suggesting a coherent application line within that venue. Beyond the original proposing team, external researchers have begun to adopt DIKW-to-DIKWP-style extensions in other application domains. For example, Wu et al. propose a DIKWP framework for circular-economy manufacturing sustainability in which the fifth element emphasizes Practice (i.e., an explicit closed-loop implementation stage) and illustrate it with empirical analysis and a case study [56]. Importantly, the DIKWP framework has been used to analyze and model the 3-No problems by mapping them onto different “spaces” of cognition. Prior research mapped imprecision (ambiguity) to the Expression Space (e.g., vague language or concepts), inconsistency to the Cognitive Space (e.g., conflicting perceptions or beliefs), and incompleteness to a gap in the Semantic Space [62,65]. By connecting and integrating these spaces, the DIKWP model enables a more robust approach to problems with incomplete data, contradictory information, or unclear descriptions.

Building on DIKWP, researchers have proposed DIKWP-TRIZ as a revolutionary extension of TRIZ that operates in the cognitive domain. Wu et al. synthesized TRIZ with DIKWP to create a framework capable of inventive problem solving in AI and artificial consciousness contexts [16]. DIKWP-TRIZ shifts TRIZ from a purely technical tool to a cognitive one: each inventive principle is reconsidered in light of transformations between DIKWP elements. This incorporation allows the methodology to explicitly tackle incomplete or inconsistent knowledge by, for example, transforming Data to Knowledge (filling knowledge gaps) or using Wisdom to resolve ethical conflicts during solution generation. A comparative analysis (see Table 1) shows that, unlike traditional TRIZ, which assumes well-defined and complete data, DIKWP-TRIZ is intent-driven and adaptive. It is designed to handle uncertain, “messy” inputs and outputs through iterative transformations and feedback, reframing problem-solving as a network of interactions among Data, Information, Knowledge, Wisdom, and Purpose rather than a linear abstraction process. Crucially, DIKWP-TRIZ emphasizes ethical and purposeful reasoning, aligning solutions with hu-

man values by leveraging the Wisdom and Purpose dimensions. These advancements lay the groundwork for a Creative AI system that can mimic human-like creativity and consciousness in solving problems.

Table 1. Comparison of traditional TRIZ with DIKWP-TRIZ.

Aspect	Traditional TRIZ	DIKWP-TRIZ
Framework	Hierarchical; focuses on technical and physical contradictions	Networked DIKWP structure; interactions among Data–Purpose
Problem Scope	Clear, well-defined technical problems (complete data)	Uncertain cognitive problems (incomplete and inconsistent data)
Innovation Method	40 fixed principles applied deterministically	25 DIKWP $\times$ DIKWP transformations; adaptive and context-driven
Handling Uncertainty	Assumes data is complete and consistent	Explicitly addresses “3-No” problems via semantic transformations and cross-dimension fixes
Outcome Alignment	No built-in ethical or purpose check	Integrates Wisdom and Purpose for ethical, goal-aligned solutions
Application Domain	Engineering design, manufacturing, etc.	AI reasoning, LLM outputs, decision-making with semantics

In summary, existing work provides the pieces required for our framework: TRIZ offers a library of creative solution patterns, DIKWP supplies a cognitive structure to embed those patterns in AI reasoning, and semantic mathematical approaches promise a formal way to maintain consistency and meaning. Our work builds upon these foundations by uniting them into a single coherent framework aimed at modeling human cognitive space and LLM expression space together.

3. Theoretical Framework for Integrating DIKWP-TRIZ with Semantic Mathematics

In this section, we present a formal framework that integrates DIKWP-TRIZ with semantic mathematics to enable Creative AI. The core idea is to treat problem-solving and expression-generation as transformations within a combined DIKWP Cognitive–Semantic Space, governed by both inventive principles (for creativity) and semantic axioms (for rigor). We first describe the DIKWP $\times$ DIKWP transformation model and then incorporate semantic mathematics to ensure each transformation is semantically sound.

3.1. DIKWP $\times$ DIKWP Transformation Flow Mechanisms for Creative AI

TRIZ operationalizes innovation by resolving contradictions, typically by mapping conflicting parameters to a set of heuristic solution strategies (the 40 Inventive Principles) and related tools such as the Contradiction Matrix and ARIZ [66]. This paradigm has been widely adopted beyond engineering, but its application can become cognitively demanding and semantically fragile when problem descriptions are incomplete, inconsistent, or imprecise, and when the “why” (value or purpose) of an innovation is not explicitly represented.

To address these limitations for creative AI and cognitive systems, Wu et al. [16] proposed DIKWP-TRIZ by integrating TRIZ with a five-dimensional cognitive semantics model: Data (D), Information (I), Knowledge (K), Wisdom (W), and Purpose (P). In DIKWP-TRIZ, the core computational idea is the DIKWP $\times$ DIKWP transformation space: the Cartesian product of the five dimensions yields a complete 5×5 index of 25 DIKWP-typed transformation categories. In an implementation, each category can be instantiated

by one or more transformation operators that map representations from one DIKWP element to another. Lower-to-higher transformations progressively enrich meaning and are particularly suitable for repairing incompleteness by adding context, abstraction, and evaluative insight; higher-to-lower transformations project goals and values into actionable requirements (rules, indicators, and data needs), which can resolve inconsistencies by constraining details under an overarching Purpose; peer-dimension transformations reduce noise, redundancy, and ambiguity, thus improving precision.

A distinctive contribution of DIKWP-TRIZ is the systematic coupling of each DIKWP $\rightarrow$ DIKWP transformation with TRIZ inventive principles. As shown in Table 2, Wu et al. construct a lookup matrix that assigns one or more TRIZ principles to every cell of the 5×5 transformation matrix. They further visualize this mapping as a heatmap, as shown in Figure 1, where each cell's intensity reflects how many TRIZ principles are applicable to that specific DIKWP transformation. In addition to serving as a creativity “injection” mechanism, the heatmap also indicates where principle overlaps may introduce uncertainty or branching, motivating a graph-search strategy that selects and iteratively refines transformation–principle sequences as problem understanding evolves [67].

Table 2. DIKWP $\times$ DIKWP transformation corresponding to the TRIZ invention principle.

From\To	D	I	K	W	P
D	1, 2, 5, 10, 12, 18, 26, 30	3, 5, 9, 17, 28, 35	6, 24	40	4, 11, 15, 29
I	10, 22	13, 17	15, 24	23, 32	16, 32
K	8, 9, 25, 27	3, 13	22, 34	15, 40	25, 31, 35
W	6, 24, 25, 35	16, 22	3, 23	15, 34	10, 15, 20, 33, 39
P	10, 19, 21, 23	6	2, 15	36, 40	1, 14, 37, 38

Note: Numbers in the table represent the numbering of the invention principles of TRIZ.

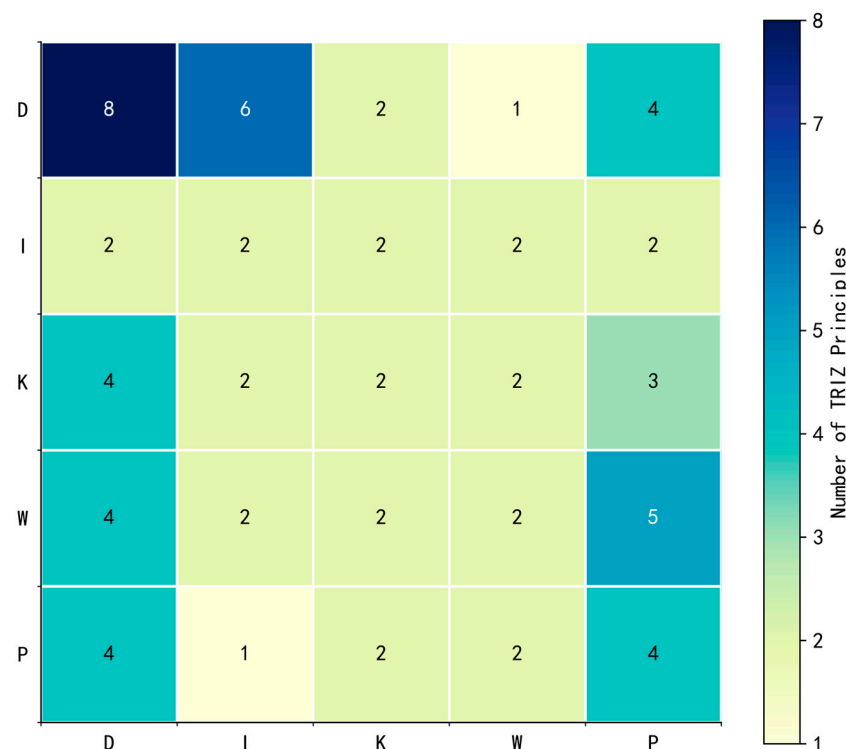


Figure 1. Heatmap of TRIZ inventive principle counts over the DIKWP $\times$ DIKWP transformation matrix.

Scope of “Coverage”. Let $L = \{D, I, K, W, P\}$ be the set of DIKWP elements. A DIKWP-typed transformation category is an ordered pair $X, Y \in L \times L$. Table 2 enumerates all 25 such categories. In this paper, the term “coverage” refers to covering this type-level transformation category space, i.e., providing a structured index and candidate TRIZ-guided heuristics for each category. This does not constitute a proof that the framework covers all possible human reasoning pathways in natural language.

In the previous version, we presented this mapping as a lookup table without explaining how such assignments should be interpreted in a cognitive/LLM setting. In this revision, we clarify that (i) the mapping is a design-time heuristic prior (not a proven optimal or complete operator set), and (ii) each TRIZ principle is used at an abstract operator level (problem-solving pattern) rather than as a literal physical mechanism.

Mapping protocol (design-time). For each cell $(X \rightarrow Y) \in L \times L$, we construct the candidate principle set by aligning the transformation intent with the abstract effect of TRIZ principles, under a fixed context ω :

- (M1) Define the transformation intent for $X \rightarrow Y$, i.e., what must change and what must be preserved (e.g., $D \rightarrow D$ emphasizes cleaning/segmentation; $P \rightarrow I$ emphasizes translating goals into information requirements).
- (M2) Abstract each TRIZ principle into one or more operator tags (e.g., segmentation, extraction, feedback, parameter adaptation, and hybridization). This abstraction is standard when transferring TRIZ from physical engineering to software/IT contexts.
- (M3) Assign a TRIZ principle to $X \rightarrow Y$ if its operator tag(s) can plausibly realize the transformation intent in the adopted representation. Multiple principles may apply; the set is intentionally permissive.
- (M4) Record the assignment as $(X \rightarrow Y, k)$ in the trace graph when the principle is actually invoked at run time, enabling later auditing and empirical ablation once a prototype exists.

Reasons why some cells list more principles than others. The number of principles in a cell reflects the breadth of plausible operator patterns for that transformation intent, not an importance weighting. For example, $D \rightarrow D$ covers several common data-level operations (segmentation, filtering, re-ordering, sampling, and format conversion), so multiple TRIZ patterns naturally match; in contrast, $P \rightarrow I$ is intentionally narrow in our current mapping, capturing a minimal “translation” from goals/constraints to information needs. We therefore treat Table 2 as an initial candidate set that can be expanded or re-weighted in future work based on expert elicitation and empirical evaluation.

Interpreting physically named principles in cognitive transformations. Some TRIZ principles originate from physical effects (e.g., “Thermal Expansion,” “Accelerated Oxidation,” and “Composite Materials”). In DIKWP-TRIZ, we apply them as abstract patterns: expansion/contraction $\leftrightarrow$ adjusting constraint ranges or search space; accelerated oxidation $\leftrightarrow$ accelerating deprecation of outdated states/purposes under new evidence; composite materials $\leftrightarrow$ hybridizing heterogeneous resources (multi-source evidence, multi-view models, or hybrid symbolic–neural representations). Appendix A (Table A2) provides a compact operator-level glossary and examples for the principles most likely to be questioned in cognitive settings.

Examples. (i) $D \rightarrow W$ with Principle 40 (“Composite Materials”): The operator is hybridization—combining heterogeneous data sources or modalities (telemetry + user constraints + domain rules) to support a higher-level insight or judgment that would not be available from any single source. (ii) $K \rightarrow W$ with Principles 15 (“Dynamicity”) and 40: Dynamicity corresponds to adaptive rule/context revision (updating a knowledge rule under changing constraints), and composite corresponds to fusing multiple knowledge fragments into a coherent wisdom-level trade-off. (iii) $P \rightarrow P$ with Principles 37/38:

Expansion corresponds to rescaling goals/constraints when the environment changes (e.g., relaxing a soft constraint while maintaining hard safety constraints), while accelerated oxidation corresponds to intentionally phasing out obsolete sub-goals when new evidence renders them invalid. These interpretations are operationalized as controlled edits to the purpose specification P_{spec} and its associated constraint set C_{ω} .

Operationally, the $DIKWP \times DIKWP$ transformation flow is implemented as modular transformation operators applied iteratively until the internal representation satisfies adequacy criteria with respect to the 3-N deficiencies. Rather than following a linear abstraction pipeline, a creative agent can dynamically navigate the DIKWP network (e.g., $D \rightarrow I \rightarrow K$ to structure understanding, $K \rightarrow W$ to assess implications, $W \rightarrow P$ to verify goal alignment, and $P \rightarrow \{K, I, D\}$ to adjust implementation details), thereby providing a structured yet flexible pathway for creative problem solving without semantic dead-ends.

3.2. Integrating Semantic Mathematics into DIKWP-TRIZ

While DIKWP-TRIZ provides structure and creativity, semantic mathematics provides formal rigor. We integrate semantic mathematics into the framework as the “glue” that ensures each DIKWP transformation preserves or enhances meaning in a logically consistent way. In practical terms, we associate each element and transformation in DIKWP with a semantic representation (such as logical propositions, conceptual graphs, or algebraic structures) that can be manipulated with mathematical rules [68]. Semantic mathematics treats semantics as both the source and target of abstraction, meaning that mathematical operations are grounded in the meaning of concepts, not just their syntactic form. This is crucial for an AI dealing with language: it ensures that when the AI transforms an idea, it does so by operating on the idea’s meaning, thereby reducing the chance of losing context or introducing contradictions.

In our framework, we formalize three interconnected representational spaces (Cognitive Space, Semantic Space, and Expression Space). Each space captures DIKWP content at a different dimension, and each DIKWP element spans all three spaces:

- **Cognitive Space:** Contains the agent’s internal mental representations (perceptions, thoughts, and intermediate reasoning states). DIKWP elements in this space correspond to the agent’s internal state (e.g., raw sensory data or personal goals).
- **Semantic Space:** Contains abstracted, language-agnostic meanings of concepts and their relations. DIKWP elements here are represented in a formal semantic model (e.g., ontologies or logical formulas capturing the content of data, information, knowledge, etc.).
- **Expression (Conceptual) Space:** Interfaces with language and symbols. DIKWP elements in this space correspond to external language and symbol expressions (words, sentences, symbols) that convey the content to the outside world.

Semantic mathematics enables rigorous mapping between these spaces. For instance, moving from the Cognitive Space to the Semantic Space involves formalizing a concept—taking a perceived or internally represented idea and assigning it a structured semantic representation (features, relations, or categories). Conversely, mapping from the Semantic Space to the Expression Space is essentially verbalizing or symbolizing the semantic representation into a natural language statement or other symbolic expression. We leverage mathematical structures (such as lattices or category-theoretic constructs) to ensure these cross-space mappings have desirable properties (e.g., being injective or surjective as needed, preserving information content in line with our axioms discussed below). By doing so, the framework is sufficient to ensure that for every cognitive entity there is a well-defined semantic representation and a corresponding expression, and that operations

on knowledge (such as combining two pieces of information) follow algebraic laws that mirror logical inference.

Example (single semantic language). Fix a context ω and an underlying formal language L_ω for semantic representations (propositional in the simplest case; first-order, description logic, or typed graph constraints in richer domains). Let $\varphi_A, \varphi_B, \varphi_C \in L_\omega$ denote formulas representing two observed items and a candidate conclusion.

$$A, B \vdash C, \quad (1)$$

In the Semantic Space, inference is written as $\{\varphi_A, \varphi_B\} \vdash_\omega \varphi_C$ (equivalently, $\Gamma \models_\omega \varphi_C$, given a background theory K_ω), meaning that φ_C is supported by φ_A and φ_B under ω . This unifies the earlier propositional and first-order presentations: propositional entailment is simply the special case where L_ω has no variables/quantifiers.

$$\forall x \forall y (P_A(x) \wedge P_B(y) \rightarrow C'(x, y)), \quad (2)$$

To promote an instance-level information conclusion to knowledge, the system may apply a generalization operator Gen_ω that produces a rule schema from observed instances. For example, from an instance derivation $\varphi_A(a) \wedge \varphi_B(b) \vdash_\omega \varphi_C = (a, b)$, the system may form a generalized rule $\forall x \forall y (P(x) \wedge Q(y) \rightarrow R(x, y))$. The Semantic Verifier checks that Gen_ω is admissible relative to K_ω and does not violate the constraints in Section 5.

When further promoting knowledge to wisdom-/purpose-aware decisions, the representation is enriched with normative or feasibility constraints (e.g., *Allowed*($\cdot$), *Safe*($\cdot$), $Cost \leq Budget$) that are evaluated under ω . Finally, the system verbalizes the validated semantic representation via lex_ω , ensuring that the expressed conclusion preserves the intended logical relations and avoids ambiguity in the adopted representation.

Integrating semantic mathematics is intended to help mitigate LLM hallucinations and logical inconsistencies by introducing explicit, checkable constraints on intermediate representations and their mappings to text. By performing consistency checks at each reasoning step, the framework can detect when a proposed inference or a generated span is unsupported under the current context or contradicts established constraints. In such cases, the system can invoke corrective transformations (e.g., dropping from a Knowledge-level assertion back to Information to retrieve evidence, or from Wisdom back to Knowledge to reassess assumptions). This feedback loop is designed to reduce error propagation and to maintain an explicit “semantic contract” between the internal reasoning state and the LLM’s outward expression.

3.3. Purpose-Driven Inference Mechanism

A distinguishing aspect of our framework is the explicit role of Purpose in inference. Traditional AI reasoning often lacks a built-in notion of “why” a solution is being pursued, whereas human cognition is almost always influenced by purpose or intent. In DIKWP-TRIZ, the Purpose dimension injects goal orientation into the problem-solving process. We formalize this via a purpose-driven inference loop: the AI maintains an explicit representation of the problem’s Purpose (which might derive from a user’s query or an assigned mission) and uses it as a guiding constraint on transformations.

Concretely, the Purpose element is represented in semantic terms (e.g., an objective function or a set of success criteria that any solution must satisfy). During each step of reasoning, the framework evaluates intermediate results against the Purpose. For example, if the Purpose is to find a solution that is safe and efficient, the Wisdom dimension (which is aware of ethical and efficiency considerations) will evaluate potential solutions or knowledge against these criteria, discarding or modifying those that conflict with the purpose.

This evaluation is done using semantic rules—e.g., a rule might state that any proposed action that causes harm violates the safety criterion (a Wisdom-dimension rule stemming from ethical knowledge). Such rules are part of the semantic math knowledge base.

Purpose-driven inference also means the system can prioritize which DIKWP transformations to apply. If the purpose clearly demands a certain type of solution (say, the user’s goal is explicitly about finding a creative design, which hints that a Knowledge $\rightarrow$ Wisdom or Wisdom $\rightarrow$ Purpose transformation is needed to introduce a novel principle), the framework can focus on those transformations rather than exhaustively exploring all. In other words, Purpose acts akin to a heuristic guiding the search through the network of transformations, pruning paths that are irrelevant to the end goal and emphasizing those likely to yield a purpose-aligned solution. This makes the problem-solving process more efficient and aligned with user expectations.

Notably, incorporating Purpose directly addresses one of the criticisms of traditional TRIZ: the neglect of human value orientation. In our framework, any solution that the AI generates must pass through the Purpose filter, which evaluates candidates against human-defined values or desired outcomes to improve alignment. This is particularly important in Creative AI for sensitive domains (medicine, law, etc.), where a “creative” solution is only useful if it respects ethical and practical constraints. By weaving Purpose into the inference loop, the framework adds explicit goal/value conditioning: the AI’s reasoning process has an internal model of “self-goal” that shapes its thoughts, somewhat analogously to how human consciousness uses intent to shape cognition.

In summary, the theoretical framework combines a network of DIKWP cognitive transformations (to cover the DIKWP-typed transformation category space and address the 3-No problems) with semantic mathematical grounding (to ensure formal consistency and meaningful mapping to expressions), all orchestrated by a purpose-driven control that aligns the process with desired outcomes.

4. Formalizing 3-No Problems in LLM Inputs and Outputs

One of the core motivations for our framework is to tackle the “3-No” problems as they manifest in LLM-based interactions. Here we formalize these issues from the perspective of input (user queries or prompts) and output (LLM responses), and describe how the DIKWP-TRIZ and semantic math framework addresses each. Understanding these problems in a structured way is crucial for designing the solution.

4.1. Definition and Formalization of 3-No Problems

Incompleteness refers to missing information or gaps in knowledge. In LLM inputs, incompleteness appears as under-specified questions or a lack of necessary context (e.g., a user asks a vague question without providing key details). The LLM thus starts with partial data. In outputs, incompleteness shows up as answers that omit important details, steps, or edge cases—the response may be correct but not comprehensive, or it might fail to finish a line of reasoning. This often happens because the model does not know certain facts or did not retrieve them, leading to incomplete solutions. Traditional LLMs have no built-in mechanism to realize something is missing; they rely on what’s present in the prompt and training data, often yielding superficially plausible but incomplete answers.

Inconsistency entails contradictions or misalignments in information. In inputs, this could be a user prompt that contains conflicting statements or a dialog history where the user changed requirements (for example, asking for X , then later saying “assume not X ”). The LLM is then faced with incoherent input data. In outputs, inconsistency is seen when the model’s response contradicts itself or factual reality—for instance, the model might make two statements in an answer that cannot both be true, or “hallucinate” a

fact that conflicts with known information. These inconsistencies arise because the LLM might locally optimize coherence in small spans of text but lacks a global truth-checking mechanism. Without an internal world model or logic rules, it may not notice when it asserts something that violates earlier statements or common knowledge.

Imprecision covers ambiguity and vagueness. Input prompts may be imprecise when they use ambiguous language, undefined terms, or open-ended requests (e.g., “Tell me about technology”—with no specific focus, this is ill-defined). The LLM then has to guess user intent. On the output side, imprecision appears as answers that are overly general, use ambiguous terms, or hedge so much that they fail to provide a clear solution. It can also mean the answer does not directly address the question (off-target), because the model was not sure what specifics were needed. Imprecise outputs often stem from the model averaging over many possible contexts or being unsure, thus giving a generic or equivocal answer. Table 3 summarizes these 3-No challenges and how our framework mitigates them.

Table 3. Manifestations of 3-No problems in LLM inputs/outputs and mitigation strategies in the proposed framework.

3-No	Input Challenge	Output Challenge	DIKWP-TRIZ Strategy
Incompleteness	Missing or insufficient data and context in user queries	Partial or under-specified answers (gaps in reasoning)	Cross-dimension enrichment: Use Data → Information conversion to supplement missing details; integrate external knowledge bases to fill gaps. The system detects missing pieces and iteratively gathers or infers data until a complete answer can be formed.
Inconsistency	Conflicting or contradictory statements in input	Self-contradictory or factually incorrect responses	Consistency checks: Cross-validate facts and claims via Knowledge and Wisdom dimensions; apply Wisdom to resolve contradictions and enforce coherence. The framework uses semantic logic to detect internal conflicts, prompting transformations (e.g., revise Knowledge using Data evidence) to eliminate inconsistencies.
Imprecision	Vague or ambiguous problem descriptions	Overly general or ambiguous answers that lack clarity	Semantic clarification: Refine and disambiguate concepts via Semantic Space transformations; utilize Purpose to focus on the specific user intent. The model will rephrase or ask questions (using DIKWP interactions) to narrow down meaning and produce a precise, context-appropriate response.

In formal terms, we incorporate these strategies into our reasoning algorithm as follows. When a user input is received, the system first parses it into the DIKWP representation in the Expression Space. At this stage, it flags potential 3-No problems:

- For incompleteness: It checks if all required DIKWP elements to answer the query are present. If certain data or information is missing, a placeholder or query is created. For example, if the question is “How do I improve system performance?” and it lacks context (what system? what metrics?), the system notes incomplete Data and Information.
- For inconsistency: It checks the input for logical contradictions or mutual exclusivity. This uses a Knowledge base of facts; any input statements that conflict (or conflict with

known facts) are marked, e.g., if the user says “I need a vegan recipe with chicken,” the system flags a contradiction between vegan and chicken.

- For imprecision: It analyzes linguistic ambiguity (multiple interpretations) by mapping terms in the query to the Semantic Space. If a term has several possible concepts (e.g., “bank” could mean river bank or financial bank), or the request is too broad, the system flags them accordingly.

Once flagged, the DIKWP-TRIZ mechanism engages to resolve these issues as part of understanding the query:

- Missing data triggers a Data → Information search, perhaps querying a knowledge base or asking the user for clarification (if interactive). This is analogous to how a person would ask follow-up questions. In our framework, an Information innovation step might be to automatically gather context (for example, retrieving relevant background info from stored data [69]).
- Contradictions in input are handled by Wisdom → Knowledge transformation: The system uses higher-order wisdom (e.g., a rule “vegan means no animal products”) to adjust or interpret the query in a consistent way (maybe by assuming the user’s priority or reinterpreting the request as a hypothetical). Alternatively, the system may split the problem into sub-problems to handle each scenario consistently, using TRIZ’s separation principles creatively to accommodate the contradiction.
- Ambiguity in input is addressed by Purpose-driven disambiguation: The system infers the likely user intent (Purpose) from whatever hints are available (user profile, conversation history, etc.) and uses that to choose an interpretation. For instance, if a user asks about “bank account” in a financial advice context, Purpose (seeking financial advice) guides the conceptual disambiguation of “bank” to the financial institution meaning. If uncertainty remains, the system can generate an Information → Data question to the user for clarification (in effect, a follow-up question) [70].

On the output side, the generation process similarly checks for 3-No problems:

- As the answer is being composed (in the Expression Space), the system monitors completeness: Have all parts of the question been addressed by some Data and Information? The DIKWP model can enforce a kind of coverage integrity, where the Cognitive Space coverage is analyzed to ensure no aspect is overlooked. If an expected component is missing, the generation is not finalized until a required DIKWP transformation step (e.g., Data → Knowledge) fills the gap.
- Consistency of the output is ensured by a final Knowledge-dimension validation. Before rendering the answer into text, the system runs a semantic consistency check: all statements intended for output are checked against each other and against a fact database. Any inconsistency triggers a correction cycle (perhaps Knowledge → Data: verifying facts, or Knowledge → Knowledge: removing contradictory statements). This is akin to a proof checker or a truth-verification pass on the draft answer.
- Precision of the output is improved by tailoring the expression in the Expression Space using Purpose and Wisdom. The Purpose dimension will strip extraneous or generic content that does not serve the user’s goal (preventing vague rambling). Wisdom will ensure the phrasing is context-appropriate and unambiguous.

By formalizing the 3-No problems in this way and integrating their detection and resolution into the DIKWP-TRIZ workflow, our framework treats these traditionally troublesome issues as just another set of solvable transformations. In effect, incompleteness, inconsistency, and imprecision become inputs to the problem-solving process, not just error conditions. This approach aligns with the idea that creative cognition often involves iteratively refining a problem statement (filling gaps, resolving contradictions, or sharp-

ening the question) before or while solving it. The next section will present the overall architecture that operationalizes these ideas, showing how an LLM-based system can be structured to include DIKWP-TRIZ modules and semantic math components.

4.2. Classification of 3-No Problem Scenarios

Not all 3-No scenarios (incomplete, inconsistent, and imprecise information) are alike—they can manifest in inputs, outputs, or both. We can classify problem types based on a 3-No input–output matrix (Table 4), which enumerates how a particular deficiency in the input can lead to various deficiencies in the output if not properly addressed. This matrix also highlights the transformation needed by the DIKWP-TRIZ framework in each case. The rows below describe each combination of input issue and output issue, along with the designated mitigation strategy:

Table 4. 3-No input–output matrix.

Input/Output	Incomplete Output	Inconsistent Output	Imprecise Output
Incomplete Input	Type 1: Gaps propagate → missing steps/details; D → I → K enrichment (retrieve/ask).	Type 2: Gap-filling assumptions → contradictions; W-check + clarify/uncertainty (I → D).	Type 3: Under-specified → generic answer; P-guided scoping (P → I) + concretize.
Inconsistent Input	Type 4: Unresolved conflict → partial coverage; K → D evidence triage + consistent subset.	Type 5: Contradictions preserved/echoed; I → K consolidation + reconcile (separate/merge).	Type 6: Conflict → hedging/vagueness; W/P arbitration + case split.
Imprecise Input	Type 7: Ambiguous intent → off-target/partial; P-guided disambiguation (ask or cover branches).	Type 8: Mixed senses → internal inconsistency; I → D concretize with examples + branch separately.	Type 9: Vagueness mirrored; K → W refinement + canonical terms.

Table 4 provides a roadmap for developers to identify the nature of a 3-No problem and apply the corresponding DIKWP transformation strategy. For example, an inconsistent input leading to an inconsistent output (Type 5) indicates a failure to unify semantics—clearly motivating the use of the semantic mathematics axioms (see Section 6) to ensure a single, consistent knowledge representation. An imprecise input leading to an incomplete output (Type 7) suggests the need for purpose-driven clarification. In all cases, the DIKWP-TRIZ framework offers a targeted method (highlighted in bold under Mitigation) to transform the issue into a resolvable form, turning problematic inputs into reliable outputs.

5. Semantic Axioms and Cognitive Transformations

To establish a rigorous theoretical foundation, we define a set of semantic axioms that underpin the framework. These axioms enforce fundamental properties (existence, uniqueness, and transitivity) in the mapping between cognitive content and expressions, thereby ensuring consistency and completeness of the AI's reasoning and outputs. We also describe how these axioms apply to the DIKWP cognitive transformations.

5.1. Axioms for Semantic Consistency and Completeness

We define three context-indexed semantic constraints over an adopted formal representation. These constraints are intended to ensure internal well-formedness and trace consistency between concept-level representations and language-level expressions under a fixed context $\omega \in \Omega$.

5.1.1. Definitions and Notation

Let Ω denote the context space. A context $\omega \in \Omega$ fixes assumptions such as domain, task, dialog history, and purpose constraints. Let C be the concept (semantic) space, consisting of canonical concept identifiers and relations used by the reasoner. Let E be the expression space, e.g., text spans or symbolic strings produced/consumed by the LLM. We define two context-indexed mapping functions: interpretation $interp_{\omega} : E \rightarrow C$ and lexicalization $lex_{\omega} : C \rightarrow E$. In an implementation, $interp_{\omega}$ can be realized by semantic parsing plus disambiguation, and lex_{ω} by controlled generation or template-based verbalization.

For a given interaction, let $E_{in} \subseteq E$ be the set of expressions appearing in the user input and the system output, and let $C_{state} \subseteq C$ be the set of concept identifiers that appear in the current reasoning state (e.g., a knowledge graph or logical theory maintained by the cognitive engine).

5.1.2. The Existence Axiom

The Existence Axiom (Axiom 1) requires that (i) every expression the system uses is interpretable as some concept in the current context, and (ii) every concept used in the reasoning trace is expressible by at least one surface form selected for that context. This is a well-definedness constraint on the adopted representation (not a claim about natural-language truth).

$$\forall e \in E_{in}, \text{ } interp_{\omega}(e) \text{ is defined,} \quad (3)$$

$$\forall c \in C_{state}, \text{ } lex_{\omega}(c) \text{ is defined.} \quad (4)$$

Checkability: In a prototype, violations of Axiom 1 can be detected whenever the parser cannot assign any concept to an output span (undefined $interp_{\omega}$) or whenever a concept node lacks any realizable label/template (undefined lex_{ω}).

5.1.3. The Uniqueness Axiom

The Uniqueness Axiom (Axiom 2) enforces single-sense interpretation and unambiguous naming under a fixed context. Specifically, after contextual disambiguation, $interp_{\omega}$ is single-valued, and lex_{ω} chooses a canonical surface form for each concept id so that distinct concept identifiers are not rendered as the same expression within the same context.

$$interp_{\omega}(e) = c_1 \wedge interp_{\omega}(e) = c_2 \Rightarrow c_1 = c_2, \quad (5)$$

$$c_1 \neq c_2 \Rightarrow lex_{\omega}(c_1) \neq lex_{\omega}(c_2), \quad (6)$$

$$interp_{\omega}(lex_{\omega}(c)) = c, \forall c \in C_{state}, \quad (7)$$

Equation (7) states a round-trip consistency requirement: after lexicalizing a concept and re-interpreting the resulting expression under the same context, the system recovers the same concept identifier. This provides an explicit, testable criterion for the soundness of the representation-level mapping in our setting.

Remark: Natural language is inherently many-to-many (synonyms and homonyms). In our framework, Axiom 2 is enforced by introducing concept identifiers (sense tags) and using context-dependent lexicalization to select (or construct) a disambiguated surface form when needed (e.g., “bank(finance)” vs. “bank(river)”).

5.1.4. The Transitivity Axiom

Let $T_{X \rightarrow Y}^{\omega}$ denote a DIKWP-typed transformation operator executed under a fixed context ω , mapping a representation at element X to a representation at element Y . We say $T_{X \rightarrow Y}^{\omega}(r)$ is defined under ω if (i) the operator is computable under ω , and (ii) it yields a well-formed, type-correct output at the target element.

The Transitivity requirement (Axiom 3) enforces closure of reachability under multi-step transformations within the same fixed context segment. Concretely, if a two-step chain $X \rightarrow Y \rightarrow Z$ is defined, then a corresponding direct transformation $X \rightarrow Z$ must be defined under the same ω . Moreover, for trace-level coherence, the direct result must agree (up to canonicalization) with the composed result.

Let $\text{canon}(\cdot)$ be a canonicalization/normalization procedure on internal representations (e.g., graph normalization or formula normalization). Then for all r and fixed ω :

$$\text{defined}(T_{X \rightarrow Y}^\omega(r)) \wedge \text{defined}(T_{Y \rightarrow Z}^\omega(T_{X \rightarrow Y}^\omega(r))) = \text{defined}(T_{X \rightarrow Z}^\omega(r)), \quad (8)$$

$$\text{canon}(T_{Y \rightarrow Z}^\omega(T_{X \rightarrow Y}^\omega(r))) = \text{canon}(T_{X \rightarrow Z}^\omega(r)), \quad (9)$$

whenever both sides are defined under ω .

Checkability. A counterexample is any r such that the two-step chain is defined under ω , but the direct mapping is not (violating Equation (8)), or such that both are defined, but their canonical forms differ (violating Equation (9)).

Context shifts. If the system updates the context from ω to ω' (e.g., new constraints are introduced), the update is explicitly recorded. Transitivity checks are performed within each fixed-context segment rather than across a context boundary without an explicit bridging rule.

Pointer. Section 5.3 provides sketch proofs showing how Axioms 1–3 are enforced and preserved as invariants of the verified reasoning loop (Algorithm 1).

Algorithm 1: DIKWP–TRIZ reasoning loop with semantic verification

Input : query q , context ω , purpose specification P_{spec} , budget B

Output : answer y , optional trace G_{trace}^*

```

1:  $S \leftarrow \text{ConceptualParser}(q, \omega)$ 
2:  $G_{\text{trace}} \leftarrow \emptyset$ 
3: for  $t = 1..B$  do
4:    $\text{issues} \leftarrow \text{Detect3No}(S, \omega)$ 
5:    $A \leftarrow \text{ProposeActions}(S, \text{issues}, P_{\text{spec}})$  //  $a = (X \rightarrow Y, \text{TRIZ } k, \text{params})$ 
6:    $\text{best} \leftarrow \text{None}$ 
7:   for  $a$  in  $A$  do
8:      $S', \Delta G \leftarrow \text{CognitiveEngine}(S, G_{\text{trace}}, a)$ 
9:      $\text{ok, report} \leftarrow \text{SemanticVerifier}(S', G_{\text{trace}} \cup \Delta G, \mathcal{C}_\omega)$ 
10:    if not ok then continue
11:     $\text{score} \leftarrow \text{PurposeScore}(S', P_{\text{spec}})$ 
12:     $\text{best} \leftarrow \text{argmax}(\text{best}, (S', G_{\text{trace}} \cup \Delta G), \text{score})$ 
13:  end for
14:  if best is None then break // no valid action under  $\mathcal{C}_\omega$ 
15:   $S, G_{\text{trace}} \leftarrow \text{best}$ 
16:  if  $\text{StopCriterion}(S, P_{\text{spec}})$  then break
17: end for
18:  $y \leftarrow \text{ExpressionSynthesizer}(S, \text{SelectTrace}(G_{\text{trace}}))$ 
19: return  $y, \text{SelectTrace}(G_{\text{trace}})$ 

```

5.2. Application to DIKWP–TRIZ Transformations

Now we examine how these axioms govern the DIKWP transformations within our framework:

- When performing a DIKWP transformation (say Information $\rightarrow$ Knowledge), the Existence axiom demands that any new piece of knowledge derived must be expressible in the Semantic and Expression Space. This means our transformation algorithms cannot introduce truly ineffable concepts—if they derive something, it must be backed by data or information that can be pointed to. For instance, if the system “has a hunch” (Wisdom) that a certain solution might work, Existence forces it to either articulate that hunch (perhaps as a hypothesis) or seek data to support it, rather than silently use it. This ties into explainability: each step’s result exists in a shareable form. Conversely, if some input information has no impact on the solution, the system should explicitly recognize that (so it does not violate Existence by leaving an input unaccounted for). The result is that every input and every intermediate is tracked and can be output if needed (no hidden state that is not representable).
- The Uniqueness axiom heavily influences the knowledge representation in the DIKWP model. For each transformation, especially those that merge or abstract ideas (like Data $\rightarrow$ Information or Knowledge $\rightarrow$ Wisdom), the system must check that it is not accidentally duplicating concepts. For example, during Information $\rightarrow$ Knowledge, if two pieces of information imply the same knowledge, the framework should combine them into one knowledge concept rather than keep two separate parallel concepts (which could later diverge and cause inconsistency). Uniqueness also requires that if a transformation creates a concept that already exists, it should unify them. In TRIZ terms, this is akin to trimming redundancies—the framework actually had a goal of reducing overlaps, which aligns with our axiom. On the expression side, uniqueness ensures that when generating the final answer, if the same concept was arrived at via two different paths, it is described once. Practically, this eliminates contradictory answers like an LLM first saying “Solution is X” and later saying “Solution might be not X” due to picking different words for the same concept; our system would realize those refer to one concept X and reconcile them.
- The Transitivity axiom ensures that cognitive chains are coherent. In DIKWP-TRIZ, one might go through a chain like Data $\rightarrow$ Information $\rightarrow$ Knowledge $\rightarrow$ Wisdom. Transitivity implies that the essential logical thread is preserved: if the original data indicated a certain outcome, the wisdom-dimension conclusion should reflect that unless intentionally overridden. If a transformation overrides something (say Wisdom rejects a Knowledge piece for ethical reasons), the system notes a break in transitivity (with justification, like “this path pruned due to ethical conflict”). Most of the time, though, transformations build on each other. Transitivity allows the framework to do multi-hop reasoning reliably: consider a case where, to solve a problem, the AI does Data $\rightarrow$ Information (find relevant facts), Information $\rightarrow$ Knowledge (derive a principle), and Knowledge $\rightarrow$ Information (apply the principle to get a specific insight). Thanks to transitivity, it should reach the same specific insight if it had, for example, directly looked up data or gone another route. This property reduces order-dependence—the solution does not arbitrarily differ if the sequence of transformations changes, as long as, logically, it covers the same ground. This is important for a robust AI; it means the reasoning graph is somewhat redundant or cross-checked.

To illustrate, let us walk through a simplified logical example with these axioms: Suppose the AI needs to conclude “C” from input facts “A” and “B.”

Existence: It ensures that if “C” is concluded, there is a sentence or formula representing “C” (e.g., “Therefore, C is true under conditions X.”). It also ensured that A and B were represented to begin with.

Uniqueness: If “A and B imply C” and that is the Knowledge, the system will not also have a separate concept “A and B therefore C” floating around; it is one unified rule. If it

phrased something differently earlier, like “If A then (if B then C)”, it would understand that this is the same as “A and B imply C” and would not treat them separately.

Transitivity: If A implies D and D with B implies C, Transitivity ensures the system can derive A and B imply C. So if one part of the reasoning was implicit, it will make it explicit. In terms of transformations, maybe $A \rightarrow \text{Information gave D}$, and then $(D,B) \rightarrow \text{Knowledge gave C}$; transitivity yields an overall view that $A,B \rightarrow \text{Knowledge yields C}$ directly, meaning the final answer can be traced straight from A and B.

These axioms are embedded in the semantic reasoner of our architecture. The reasoner uses them as constraints:

- It will refuse to accept a new piece of knowledge that it cannot attach to some expression (flagging a potential violation of Existence).
- It will run a unification algorithm to enforce Uniqueness whenever the Knowledge graph is updated (merging nodes that are essentially the same concept, preventing duplicates).
- It will apply forward-chaining and backward-chaining inferences to respect Transitivity (similar to how a Prolog engine or rule-based system would derive all consequences, to avoid missing a logical link).

By adhering to these axioms, the framework improves expression-space consistency and completeness. The answers given by the AI are consistent (no internal contradictions or unexplained leaps) and complete (covering the question’s aspects with nothing critical omitted), directly addressing the common failings of typical LLM responses. The axioms also help with maintaining the “semantic contract” between the human cognitive model and the LLM expression: they formalize the expectation that the meaning stays aligned at every step. This axiomatic approach is in line with the call for “fine semantic mathematics to establish seamless connections between Cognitive Space, Semantic Space and Expression Space”, as noted by previous research, which aims to minimize ambiguity and uncertainty in AI reasoning.

In conclusion, the semantic axioms act as the theoretical foundation that makes our integration of DIKWP-TRIZ with LLMs robust. They ensure that the cognitive transformations yield outputs that are logically sound and faithfully represent the AI’s internal reasoning. Next, we discuss the broader implications: how does this framework empower human-like creativity and purposeful behavior in AI, and what new capabilities or research avenues does it open?

5.3. Sketch Proofs and Invariant Preservation

A reviewer noted that, although the axioms in Section 5.1 are now explicitly formulated and falsifiable, the manuscript should, at least, also provide sketch proofs that these axioms hold within the proposed framework. We address this by framing the reference architecture (Section 6) as a design-by-contract system: the axioms are enforced as invariants maintained by the Semantic Verifier inside Algorithm 1. Any candidate update that violates an axiom is rejected and triggers corrective transformations.

5.3.1. Minimal Contracts (Implementation Assumptions Checked by the Verifier)

Fix a context segment ω . Let S_t^ω be the DIKWP state at iteration t , and let $C_{state}(S_t^\omega) \subseteq C$ denote the set of concept identifiers currently present in the reasoning state. We assume the following minimal contracts, which can be realized by standard semantic parsing/linking, controlled lexicalization, and canonicalization/unification procedures:

- (C1) Total interpretation of in-scope expressions. For every expression $e \in E_{in}$, $interp_\omega(e)$ is defined (possibly mapping to an explicit “unknown/placeholder” concept with provenance).

- (C2) Total lexicalization of active concepts. For every concept $c \in C_{state}(S_t^\omega)$, $lex_\omega(c)$ is defined (using controlled fallback labels when needed).
- (C3) Contextual disambiguation and canonical naming. Under fixed ω , $interp_\omega$ is single-valued after disambiguation, and lex_ω is injective over active concept identifiers (using explicit sense tags if necessary).
- (C4) Canonicalization and unification. $canon(\cdot)$ yields a stable normal form for the chosen representation, and a unification step merges duplicated concepts that are equivalent under the adopted equivalence relation (e.g., same id or same canonical signature).
- (C5) Operator admissibility and closure. Each executed operator $T_{X \rightarrow Y}^\omega$ is well-typed and returns a well-formed output. For any invoked two-step chain $X \rightarrow Y \rightarrow Z$, either (i) a corresponding direct operator $T_{X \rightarrow Z}^\omega$ is available, or (ii) the system defines a direct transformation instance as the canonicalized composition, ensuring Equations (8) and (9) for that instance.

These contracts are not merely assumptions: they are precisely what the Semantic Verifier checks, via the constraint set C_ω , before accepting any update in Algorithm 1.

5.3.2. Lemmas (Axioms Hold for Any Verified State)

Lemma 1. (Existence). *Under (C1) and (C2), any state accepted by the Semantic Verifier satisfies Axiom 1 (Equations (3) and (4)).*

Proof sketch. (C1) ensures $interp_\omega(e)$ is defined for all $e \in E_{in}$, and (C2) ensures $lex_\omega(c)$ is defined for all active c . The verifier rejects states violating either condition.

Lemma 2. (Contextual Uniqueness). *Under (C3) and (C4), any accepted state satisfies Axiom 2 (Equations (5)–(7)).*

Proof sketch. (C3) makes $interp_\omega(e)$ single-valued and enforces injective canonical naming via lex_ω . (C4) prevents duplicate concepts that could otherwise cause naming collisions. The verifier tests and rejects violations (including failures of the round-trip condition in Equation (7)).

Lemma 3. (Transitivity/Compositional Trace Consistency). *Under (C4) and (C5), any accepted transformation step satisfies Axiom 3 (Equations (8) and (9)) within the fixed context segment ω .*

Proof sketch. By (C5), whenever a two-step chain is invoked and defined, the framework either provides a direct operator or defines the direct instance as the canonicalized composition. Canonicalization stability (C4) enables equality checking for Equation (9). Any counterexample is rejected by the verifier.

5.3.3. Proposition (Invariant Preservation in Algorithm 1)

Proposition 1. (Axioms 1–3 are loop invariants). *Assume the initial parsed state S_0^ω satisfies (C1)–(C3). In Algorithm 1, if each committed update passes the Semantic Verifier under C_ω , then S_t^ω satisfies Axioms 1–3 for all iterations t until termination. Consequently, the final answer synthesized from S_*^ω is generated from an axiom-compliant internal representation.*

Proof sketch. Induction on t . Base: S_0^ω satisfies Axioms 1 and 2 by Lemmas 1 and 2. Step: Any candidate S_{t+1}^ω is committed only if the verifier accepts it; by Lemmas 1–3, acceptance implies Axioms 1–3 hold. Therefore, the axioms are preserved across the loop.

Scope note. These sketches establish enforceable well-formedness and trace-consistency of the concept-expression mapping under fixed context segments. They do not claim global truth soundness/completeness for open-world natural-language statements.

6. Reference Architecture and Implementation Blueprint for a DIKWP-TRIZ and Semantic Mathematics Enhanced LLM

6.1. Reference Architecture Overview

To operationalize the theoretical framework, we propose a reference architecture that integrates DIKWP elements and $\text{DIKWP} \times \text{DIKWP}$ transformation categories into an LLM-centered workflow. This section is intended as an implementable blueprint (module decomposition, intermediate representations, and a reference execution loop), rather than a claim of a fully deployed prototype. Figure 2 provides a high-level overview of the proposed DIKWP-TRIZ + semantic-mathematics pipeline.

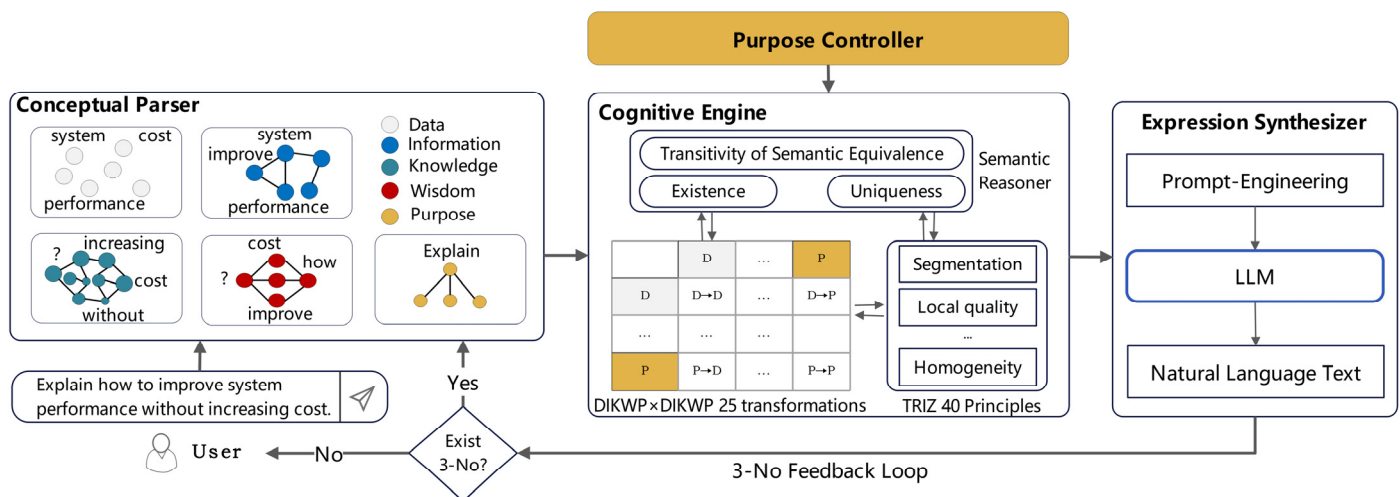


Figure 2. Conceptual overview of the reference architecture (modules and data flow).

The model involves an iterative loop between cognitive processing and expression generation, all guided by semantic checks. In the diagram, a user query enters at the left and is decomposed into DIKWP elements in the Expression Space. These elements undergo transformations in the Cognitive Space (center) driven by TRIZ's 40 principles and semantic mathematics. The Purpose module (top) continuously guides the process, filtering and directing transformations according to the goal. The LLM's expression space (right) receives the finalized conceptual answer, which is then output as natural language. The process is a closed-loop: if the output is evaluated (by either the system or user feedback) to have 3-No problems, the cycle repeats with adjustments.

6.1.1. Intermediate Representations and Data Structures

To make the architecture implementable, we specify the intermediate representations exchanged between modules under a fixed context $\omega \in \Omega$.

- Context ω : A context record capturing domain, dialog state, time, and user constraints that index disambiguation and validity checks.
- DIKWP state $S^\omega = \langle D, I, K, W, P \rangle$: A typed frame in which each element holds a set (or list) of items with provenance.
- Item schema: Each item is represented as a tuple (id, content, element, confidence, provenance, and constraints), where provenance links to source evidence or prior transformations.
- Trace graph G_{trace} : A directed graph storing transformation applications as edges (transform type $X \rightarrow Y$, TRIZ principle id, context ω , timestamp). This supports auditability and explainability.

- Constraint set $\mathcal{C}_\omega$: A set of context-indexed consistency constraints derived from Section 5 (Existence/Expressibility, Contextual Uniqueness, and Transitivity (Compositional Trace Consistency)) that are checkable on S^ω and G_{trace} .

6.1.2. Module Interfaces

We define the modules in Figure 2 as interface-level components. In an implementation, each module may be realized by prompting an LLM, calling external tools, and using symbolic reasoning engines; the key requirement is that the inputs/outputs match the following interface contracts.

Conceptual Parser. Input: User query q (text), context ω .

Output: Initial DIKWP state S^{ω_0} , entity/constraint mentions, and a mapping between surface spans in q and concept identifiers in S^{ω_0} .

Notes: Can be implemented as (i) LLM-structured extraction (JSON schema) or (ii) a semantic parser + ontology linker.

Cognitive Engine. Input: Current DIKWP state S^{ω_i} , trace graph G_{trace} , and candidate transformation action $a = (X \rightarrow Y, TRIZ\ principle\ k, parameters)$.

Output: Transformed state $S^{\omega_{i+1}}$ and trace updates ΔG_{trace} .

Notes: Each DIKWP transformation category $(X, Y) \in L \times L$ is instantiated as a transformation operator family $T_{X \rightarrow Y}^\omega(\cdot)$ guided by the selected TRIZ principle(s).

Semantic Verifier (Consistency Checker). Input: Candidate state S_{t+1}^ω , trace G_{trace} , and constraint set $\mathcal{C}_\omega$.

Output: Pass/fail flag plus a minimal violation report (which constraint failed and which items caused it).

Notes: Can be implemented using rule checking, unification, description-logic reasoning, or lightweight constraint solving; the paper remains representation-agnostic.

Purpose Controller (Planner and Scorer). Input: Purpose specification P_{spec} (goal, hard constraints, and soft preferences), current state S_t^ω , and verifier feedback.

Output: Next action selection a_t and a stopping decision.

Notes: Can be implemented as a heuristic search over transformation actions, with a purpose score function and a budget for exploration.

Expression Synthesizer. Input: Final validated state S_*^ω , selected supporting trace subgraph G_{trace}^* .

Output: Natural-language answer y , optionally with an explanation trace (references to D/I/K/W/P items).

Notes: Typically implemented with an LLM conditioned on the structured representation (e.g., constrained prompting). In particular, $\mathcal{C}_\omega$ includes the checkable conditions corresponding to Axioms 1–3, and Section 5.3 provides sketch proofs that these constraints are preserved as invariants of Algorithm 1 for accepted states.

6.1.3. Reference Pipeline Algorithm

Algorithm 1 sketches an execution loop consistent with the module interfaces above. It is intentionally implementation-agnostic (it can be realized as an agent loop or a pipeline), while remaining falsifiable under a fixed context ω via the Semantic Verifier.

Finally, the proposed architecture is modular, meaning it could be implemented by combining existing technologies: a knowledge graph and reasoner for the semantic part, a fine-tuned LLM for parsing and generation, and possibly rule-based systems for certain DIKWP transformations [71–73]. The novelty lies in how they are orchestrated under the DIKWP-TRIZ philosophy.

In terms of systems design, this could be realized as a pipeline or an agent with multiple experts:

- A Planner (controlled by Purpose) that decides which transformation (expert) to invoke next.
- Multiple Expert Modules (one per DIKWP transform or group of transforms) that carry out specific cognitive tasks and report back results and confidence.
- A Global Knowledge Base (including semantic rules, ontology, and facts) accessible to all modules for consistency checks.

The LLM acting in dual roles: As a semantic parser (input understanding) and natural language generator (output expression), possibly augmented with few-shot prompts or chain-of-thought to explain its reasoning as it goes (which can be fed into the semantic checker).

Security and control-wise, the architecture also allows transparency: because intermediate reasoning is represented in semantic terms, we can log and inspect the “thought process” of the AI (something often lacking in pure LLM systems). This makes it easier to debug or refine the system and to ensure it is following the intended axioms and principles.

6.2. Illustrative Comparison and Evaluation Protocol

A conventional LLM and a DIKWP-TRIZ- and semantic-mathematics-enhanced LLM exhibit fundamentally different reasoning behaviors. The proposed approach introduces a structured cognitive pipeline based on DIKWP, leverages TRIZ principles to guide creative and contradiction-resolving transformations, and applies semantic mathematics to enforce logical constraints and consistency checks throughout the reasoning process. Together, these mechanisms aim to improve reasoning reliability, goal alignment, and interpretability. The following sections compare the two approaches across key dimensions and summarize the differences using illustrative figures.

6.2.1. Knowledge Representation

- Standard LLM: Knowledge is implicit and distributed across model parameters, without an explicit, inspectable knowledge base [69]. As a result, the model may rely on pattern completion when facing missing or uncertain facts, which can lead to factual inaccuracies or unsupported assertions [5]. The absence of transparent provenance also makes it difficult to verify where a claim originates or to systematically correct errors.
- DIKWP-TRIZ and Semantic Mathematics Enhanced LLM: Knowledge is represented explicitly in a multi-element form aligned with DIKWP semantics, and can be linked to structured repositories such as knowledge graphs, ontologies, or curated databases. Factual content can be retrieved on demand and validated against semantic constraints. This separation between stored knowledge and language generation improves traceability and enables systematic verification, thereby reducing unsupported generations and improving factual robustness.

6.2.2. Reasoning Process

- Standard LLM: Reasoning is typically realized as an end-to-end generation process that maps an input prompt directly to an output response. Intermediate reasoning states are not inherently structured or externally verifiable. For multi-step tasks, errors introduced early may propagate into the final answer, and the lack of explicit checkpoints makes it difficult to detect contradictions or missing steps before the response is produced [74,75].
- DIKWP-TRIZ- and Semantic-Mathematics-Enhanced LLM: Reasoning is structured and iterative, progressing through DIKWP stages with explicit intermediate artifacts (e.g., extracted data items, inferred relations, generalized knowledge statements, value-aware judgments, and purpose-aligned decisions). A semantic mathematics

component performs constraint-based validation (e.g., consistency checking and compatibility with predefined axioms) at each stage. When contradictions or trade-offs are encountered, TRIZ principles provide transformation operators that support creative reframing and resolution (e.g., decomposing conflicts, separating constraints, or restructuring the problem). This staged workflow functions as a semantic supervision loop, reducing error accumulation and improving coherence in complex reasoning tasks.

6.2.3. Context and Purpose

- **Standard LLM:** The model primarily responds to the immediate prompt and local context, with no explicit internal representation of a persistent long-term objective. In extended interactions, this can result in topic drift, inconsistent decisions, or weakened adherence to constraints [76], especially when the user's intent must be maintained across multiple turns [77].
- **DIKWP-TRIZ and Semantic-Mathematics-Enhanced LLM:** The Purpose dimension provides a persistent objective that guides the selection of relevant information, prioritizes memory and context, and filters candidate reasoning paths. Purpose-driven control enables planning-like behavior: intermediate conclusions can be evaluated against the goal and constraints, and revised when misalignment is detected. This mechanism improves long-horizon coherence in multi-turn dialog by reducing drift and enforcing goal-consistent reasoning across turns.

6.2.4. Handling “3-No” Inputs

- **Standard LLM:** A standard LLM typically lacks explicit mechanisms for detecting and resolving incomplete, inconsistent, or imprecise inputs (the “3-No” issues) [70]. When a user prompt omits key details, contains internal contradictions, or is underspecified, the model often relies on implicit priors learned during training to fill gaps. In practice, it may gloss over ambiguity or introduce unstated assumptions without explicitly signaling uncertainty [78]. This behavior can yield vague or internally conflicting outputs, particularly when the prompt includes hidden inconsistencies that remain unrecognized [79]. As a result, ambiguity may be propagated rather than resolved, increasing the likelihood of logical errors or conflicts with earlier dialog turns.
- **DIKWP-TRIZ and Semantic-Mathematics-Enhanced LLM:** The proposed approach explicitly manages “3-No” issues through structured reasoning and constraint-aware verification. At the Data → Information stage, the system can identify missing entities, underspecified constraints, or unclear references and respond by initiating targeted clarification, retrieving supporting facts, or constructing multiple plausible interpretations under explicit constraints. During Information → Knowledge conversion, the system performs consistency checks to detect conflicts among facts, assumptions, and constraints; when contradictions emerge, it can invoke TRIZ-guided transformations to restructure the problem or separate competing requirements. Imprecision is further addressed at the Wisdom and Purpose dimensions by refining intent, prioritizing constraints, and enforcing goal-consistent interpretations. Throughout this pipeline, semantic mathematics provides constraint-based validation to reduce uncontrolled assumption propagation. Consequently, the system either resolves uncertainty via structured transformations or makes uncertainty explicit, improving robustness under underspecified or contradictory queries.

6.2.5. Alignment and Values Integration

- **Standard LLM:** In many standard settings, alignment with human values and safety requirements is introduced primarily through post hoc techniques such as preference-

based fine-tuning and output filtering [80,81]. While these methods can reduce undesirable responses, ethical and value-related constraints are not necessarily represented as first-class elements within the model's internal reasoning process. As a result, responses may be technically plausible yet contextually inappropriate or insufficiently sensitive to user intent and constraints, especially when the prompt does not explicitly specify value-based requirements.

- **DIKWP-TRIZ and Semantic-Mathematics-Enhanced LLM:** The proposed model integrates value-sensitive reasoning within its cognitive pipeline. The Wisdom dimension explicitly evaluates candidate conclusions and actions under normative constraints (e.g., safety, responsibility, and fairness) alongside task objectives. Instead of applying alignment only as a final filter, the model checks whether intermediate outcomes are not only correct with respect to knowledge, but also appropriate under value constraints and purpose requirements. Semantic mathematics supports this process by formalizing policy-like constraints as verifiable conditions, while TRIZ-guided transformations help search for alternatives that satisfy both technical goals and normative boundaries. This design encourages responses that are goal-aligned, context-sensitive, and responsibly constrained by construction.

6.2.6. Explainability and Transparency

- **Standard LLM:** Standard LLMs provide limited transparency because the internal decision process is distributed across high-dimensional parameters and does not naturally yield an inspectable reasoning trace [82]. When asked to justify an answer, the model may generate a plausible post hoc explanation, but it is not guaranteed to correspond to verifiable intermediate reasoning states [83]. This opacity complicates debugging, auditing, and error attribution, particularly in settings that require traceability and governance [84].
- **DIKWP-TRIZ and Semantic-Mathematics-Enhanced LLM:** The proposed system is designed to improve explainability by producing explicit intermediate artifacts at each DIKWP element. For example, it can expose: (i) Data elements (entities, constraints, and evidence candidates), (ii) Information (structured interpretations and relations), (iii) Knowledge (generalizations, rules, or causal links), (iv) Wisdom (value-aware assessments and trade-offs), and (v) Purpose (goal definitions and acceptance criteria). These intermediate results form an auditable reasoning trace that can be inspected to determine where errors or misalignments originate. In addition, semantic mathematics enables explicit constraint checks whose outcomes can be logged, supporting systematic verification and targeted correction at the stage where a violation is detected. This structured trace improves interpretability and facilitates debugging, governance, and trust calibration.

Overall, the DIKWP-TRIZ and Semantic-Mathematics-Enhanced LLM can be viewed as a cognitive-style agent with structured internal representations and verifiable intermediate states, whereas a standard LLM primarily operates as an end-to-end language predictor. The comparative analysis suggests that:

- **Factual Reliability:** Explicit knowledge representations and semantic validation are intended to help reduce unsupported claims and to improve verifiability, compared with purely implicit parameter-based recall.
- **Deep Multi-Step Reasoning:** Structured reasoning with intermediate checks is intended to help limit error accumulation and to improve coherence on multi-step tasks.
- **Context and Goal Persistence:** Purpose-driven control improves long-horizon consistency in multi-turn interactions by prioritizing goal-relevant information and filtering candidate reasoning paths.

- Robustness to Ambiguity (“3-No”): Structured detection and resolution of incompleteness, inconsistency, and imprecision reduces uncontrolled assumption propagation and encourages explicit clarification when needed.
- Value-Aware Reasoning: Integrating value constraints within the Wisdom and Purpose stages enables responsible decision-making during inference rather than relying solely on post hoc filtering.
- Transparency: DIKWP element intermediate artifacts provide an auditable reasoning trace that supports inspection, debugging, and governance.

These differences are expected to translate into measurable improvements on benchmarks involving multi-turn dialog, constraint satisfaction, and multi-step reasoning. While standard LLMs remain effective for fluent natural-language generation and straightforward tasks, the proposed framework shifts the reasoning paradigm toward a more deliberative, purpose-guided, and semantically constrained process, with improved reliability and interpretability for complex scenarios.

7. Discussion

1. Human-Like Creativity and Innovation.

Integrating TRIZ into the DIKWP cognitive framework introduces a systematic approach to creativity, enabling the AI to generate innovative solutions in a systematic manner. TRIZ’s 40 inventive principles provide generalizable patterns for resolving contradictions and reusing ideas across domains. By leveraging these principles, the framework can propose non-obvious solutions to complex problems (for example, using dynamically reconfigurable traffic lanes to reduce congestion by applying TRIZ’s “Dynamism” principle). These creative leaps are not random; each is generated deliberately and then vetted through the semantic mathematics module. The semantic module checks consistency with known constraints and domain knowledge, ensuring that novel ideas remain logically sound. This combination allows the AI to function akin to human creative processes—brainstorming bold ideas and then testing them systematically. Compared to conventional LLMs that only reproduce patterns seen in training data, our TRIZ-enhanced framework can systematically generate candidates and refine them in a traceable way [85].

2. Cognitive Completeness and Explainability.

The DIKWP-TRIZ system encompasses the full spectrum of cognition from raw Data to higher-order Wisdom and Purpose, which is a step toward cognitive completeness. In practice, this means the AI can handle detailed facts and abstract goals within one system. It can translate a vague higher-order objective into concrete sub-tasks and data requirements, bridging the gap between pattern-based learning and explicit logical reasoning. If asked why it suggested a particular solution, the system can justify it in terms of higher-order goals (Purpose and Wisdom), domain-specific reasoning (Knowledge), and supporting evidence (Data and Information). Such a structured explanation builds transparency and user trust, surpassing the often opaque answers of purely neural systems. The framework also explicitly handles scenarios of incomplete or inconsistent information, addressing common “3-No” problems (e.g., no available data, no existing knowledge). Rather than guessing or producing incorrect answers, the system identifies gaps in knowledge and either seeks additional information or openly acknowledges uncertainty. In effect, the system maintains an explicit representation of uncertainty (i.e., what it does not know), which can be viewed as a rudimentary form of metacognitive self-monitoring.

3. Purposeful and Ethical AI Behavior.

Introducing an explicit Purpose dimension into the cognitive architecture has direct benefits for alignment and ethical behavior, as it ensures that intermediate steps and final

solutions remain aligned with the user's goals (Purpose) and higher-order human values (Wisdom). This focus reduces irrelevant or tangential output, since content that does not serve the specified purpose is filtered out. Additionally, the Purpose and Wisdom dimensions act as an ethical compass. If a user request conflicts with fundamental ethical standards (for example, a potentially harmful action), the system will detect it and either adjust the solution or respectfully refuse to comply. This built-in alignment is more robust than post hoc filtering, because ethical considerations inform the reasoning process from the start. The framework also supports a controlled form of AI autonomy: the system can set its own sub-goals in pursuit of the user's objective. However, this autonomy is strictly bounded by the global Purpose and Wisdom, preventing the AI from taking any extreme or unapproved measures. This design is an important step toward AI that not only reasons effectively but also stays aligned with human intentions and values—a key requirement for safe, trustworthy AI.

4. Limitations and Future Work.

Notwithstanding these advantages, the proposed framework has certain challenges. The complexity of integrating symbolic reasoning (semantic mathematics and logical axioms) with neural language models can result in performance bottlenecks. Real-time cooperation between the LLM and the reasoning module needs optimization, for example, through heuristic guidance or caching intermediate results. Maintaining an extensive and up-to-date knowledge base or ontology for the semantic module is also non-trivial; the system's effectiveness depends on high-quality structured knowledge, which may be difficult to scale across domains [86]. Furthermore, strict enforcement of reasoning axioms (such as always ensuring uniqueness or consistency) may sometimes need to be relaxed. In creative or narrative tasks, for example, a degree of ambiguity or redundancy might be desirable; this means the system should learn to relax certain constraints when contextually appropriate. Adapting the DIKWP-TRIZ framework to handle multiple modalities (e.g., incorporating visual or sensory data at the Data dimension) is another promising direction, though it requires coordinating different data types with the language-based reasoning. Evaluating such a broad system will also require new metrics beyond standard accuracy, including measures of creativity, consistency, ethical alignment, and explanatory quality. Future work may expand the “3-No” approach to address additional types of knowledge deficiencies (such as lack of context or timeliness), ensuring the AI can handle an even broader range of uncertainties. Another important direction is to incorporate learning mechanisms, such as using reinforcement learning to refine how the system applies TRIZ principles and logical checks, or fine-tuning the LLM to better cooperate with the reasoner. Finally, the transparent decision process of this architecture opens opportunities for human-AI collaboration. Users or domain experts might intervene at various cognitive elements (for example, adjusting the Purpose or adding a Wisdom rule) to guide the system, making it a versatile assistant in complex problem-solving scenarios.

8. Conclusions

In summary, this paper presents a conceptual framework that integrates DIKWP-TRIZ with a semantic-mathematics layer to bridge an internal cognitive representation and the external expression space of LLMs. The main contribution is a DIKWP-typed transformation scaffold (the 5×5 DIKWP $\times$ DIKWP matrix) together with context-indexed traceability and three checkable consistency constraints (Existence, Contextual Uniqueness, and Transitivity). We further provide a reference architecture that specifies module interfaces, intermediate representations, and a verifiable execution loop under a fixed context ω . The framework is intended to mitigate uncontrolled assumption propagation under the “3-No” conditions by making intermediate states explicit, auditable, and constraint-checked.

However, the manuscript does not claim a deployed prototype, benchmark-validated performance gains, or global soundness/completeness for natural-language truth; the proposed benefits should be regarded as design goals and hypotheses to be tested empirically. Future work will implement a prototype, define concrete operator instantiations for key transformation categories, and conduct reproducible evaluations with clearly specified baselines, datasets, metrics, and statistical reporting. Another priority is to substantiate and refine the TRIZ ↔ DIKWP mappings via documented derivations, expert elicitation, and ablation studies. We hope this blueprint enables systematic engineering and evaluation of purpose-guided, semantically constrained LLM systems.

Author Contributions: Conceptualization, Z.G. and Y.D.; methodology, Z.G. and Y.D.; formal analysis, Z.G.; writing—original draft, Z.G.; writing—review and editing, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 72462016; the Hainan Province Health Science and Technology Innovation Joint Program, grant number WSJK2024QN025; and the Hainan Province Key R&D Program, grant numbers ZDYF2022GXJS007 and ZDYF2022GXJS010.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included within the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. TRIZ 40 inventive principles.

Principle No.	TRIZ Inventive Principle
1	Segmentation
2	Extraction (Taking out)
3	Local Quality
4	Asymmetry
5	Consolidation (Merging)
6	Universality
7	Nesting (Matrioshka)
8	Counterweight (Anti-weight)
9	Prior Counteraction (Preliminary anti-action)
10	Prior Action (Preliminary action)
11	Cushion in Advance (Beforehand cushioning)
12	Equipotentiality
13	Do It in Reverse (The other way round)
14	Spheroidality (Curvature)
15	Dynamicity (Dynamics)
16	Partial or Excessive Action
17	Transition Into a New Dimension (Another dimension)
18	Mechanical Vibration
19	Periodic Action
20	Continuity of Useful Action
21	Rushing Through (Skipping)
22	Convert Harm Into Benefit (Blessing in disguise)
23	Feedback

Table A1. Cont.

Principle No.	TRIZ Inventive Principle
24	Mediator (Intermediary)
25	Self-service
26	Copying
27	Dispose (Cheap short-lived objects)
28	Replacement of Mechanical System (Mechanics substitution)
29	Pneumatic or Hydraulic Constructions (Pneumatics and hydraulics)
30	Flexible Membranes or Thin Films (Flexible shells and thin films)
31	Porous Material (Porous materials)
32	Changing the Color (Color changes)
33	Homogeneity
34	Rejecting and Regenerating Parts (Discarding and recovering)
35	Transformation of Properties (Parameter changes)
36	Phase Transition (Phase transitions)
37	Thermal Expansion
38	Accelerated Oxidation (Strong oxidants)
39	Inert Environment (Inert atmosphere)
40	Composite Materials

Table A2. Operator-level interpretation of selected TRIZ principles used in cognitive/LLM transformations.

TRIZ Principle (No., Name)	Operator-Level Interpretation in DIKWP/LLM Setting	Example Instantiation (Cell-Level)
40, Composite Materials	Hybridization/composition: Combine heterogeneous resources (multi-source evidence, multi-view models, symbolic + neural components) into a single representation.	D → W: fuse heterogeneous data streams into a higher-level insight; K → W: merge multiple knowledge fragments into a trade-off judgment.
37, Thermal Expansion	Constraint-range expansion/contraction: Adapt the feasible region or search space by relaxing/strengthening soft constraints while preserving hard constraints.	P → P: Rescale goals/constraints under changing context; P → I: Expand information requirements when purpose broadens.
38, Accelerated Oxidation	Accelerated deprecation: Intentionally phase out obsolete assumptions/sub-goals when new evidence arrives; increase the update rate of a stale state.	P → P: retire outdated sub-goals; K → K: rapidly revise a rule base when contradictions are detected.
15, Dynamicity	Adaptive structure: Allow representations and rules to change with context; support dynamic reconfiguration instead of fixed pipelines.	K → W: adapt value-aware judgment as constraints change; I → K: update generalizations when new cases appear.
6, Universality	General-purpose representation: Translate a goal into a reusable information requirement or shared intermediate schema.	P → I: derive a unified information schema from a purpose specification to avoid ad hoc interpretation.
1, Segmentation	Decomposition: Break a problem/state into separable subparts for localized repair, retrieval, or reasoning.	D → D/I → I: segment inputs; K → K: split conflicting rules into cases; P → P: decompose a global objective into sub-goals.

References

1. Peykani, P.; Ramezanlou, F.; Tanasescu, C.; Ghanidel, S. Large language models: A structured taxonomy and review of challenges, limitations, solutions, and future directions. *Appl. Sci.* **2025**, *15*, 8103. [CrossRef]

2. Shen, S.; Logeswaran, L.; Lee, M.; Lee, H.; Poria, S.; Mihalcea, R. Understanding the capabilities and limitations of large language models for cultural commonsense. *arXiv* **2024**, arXiv:2405.04655. [CrossRef]

3. Suzuki, Y.; Banaei-Kashani, F. Universe of Thoughts: Enabling Creative Reasoning with Large Language Models. *arXiv* **2025**, arXiv:2511.20471. [[CrossRef](#)]
4. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 610–623.
5. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [[CrossRef](#)]
6. Maatouk, A.; Piovesan, N.; Ayed, F.; De Domenico, A.; Debbah, M. Large language models for telecom: Forthcoming impact on the industry. *IEEE Commun. Mag.* **2024**, *63*, 62–68. [[CrossRef](#)]
7. Shanahan, M. Talking about large language models. *Commun. ACM* **2024**, *67*, 68–79. [[CrossRef](#)]
8. Urlana, A.; Kumar, C.V.; Singh, A.K.; Garlapati, B.M.; Chalamala, S.R.; Mishra, R. LLMs with Industrial Lens: Deciphering the Challenges and Prospects—A Survey. *arXiv* **2024**, arXiv:2402.14558.
9. Kim, J.; Podlasek, A.; Shidara, K.; Liu, F.; Alaa, A.; Bernardo, D. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *Sci. Rep.* **2025**, *15*, 39426. [[CrossRef](#)]
10. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [[CrossRef](#)]
11. Lake, B.M.; Ullman, T.D.; Tenenbaum, J.B.; Gershman, S.J. Building machines that learn and think like people. *Behav. Brain Sci.* **2017**, *40*, e253. [[CrossRef](#)] [[PubMed](#)]
12. Pearl, J. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv* **2018**, arXiv:1801.04016. [[CrossRef](#)]
13. Coecke, B.; Sadrzadeh, M.; Clark, S. Mathematical foundations for a compositional distributional model of meaning. *arXiv* **2010**, arXiv:1003.4394. [[CrossRef](#)]
14. Al'tshuller, G.S. *The Innovation Algorithm: TRIZ, Systematic Innovation and Technical Creativity*; Technical Innovation Center, Inc.: Worcester, MA, USA, 1999.
15. Altshuller, G.S. *Creativity as an Exact Science*; CRC Press: Boca Raton, FL, USA, 1984.
16. Wu, K.; Duan, Y. DIKWP-TRIZ: A Revolution on Traditional TRIZ Towards Invention for Artificial Consciousness. *Appl. Sci.* **2024**, *14*, 10865. [[CrossRef](#)]
17. Mei, Y.; Duan, Y. The DIKWP (Data, Information, Knowledge, Wisdom, Purpose) Revolution: A New Horizon in Medical Dispute Resolution. *Appl. Sci.* **2024**, *14*, 3994. [[CrossRef](#)]
18. Sojka, V.; Lepsik, P. Tools of Theory of Inventive Problem Solving Used for Process Improvement—A Systematic Literature Review. *Processes* **2025**, *13*, 226. [[CrossRef](#)]
19. Ilevbare, I.M.; Probert, D.; Phaal, R. A review of TRIZ, and its benefits and challenges in practice. *Technovation* **2013**, *33*, 30–37. [[CrossRef](#)]
20. Chai, K.-H.; Zhang, J.; Tan, K.-C. A TRIZ-based method for new service design. *J. Serv. Res.* **2005**, *8*, 48–66. [[CrossRef](#)]
21. Beckmann, H. Method for transferring the 40 inventive principles to information technology and software. *Procedia Eng.* **2015**, *131*, 993–1001. [[CrossRef](#)]
22. Yan, W.; Zanni-Merk, C.; Cavallucci, D.; Collet, P. An ontology-based approach for inventive problem solving. *Eng. Appl. Artif. Intell.* **2014**, *27*, 175–190. [[CrossRef](#)]
23. Ghane, M.; Ang, M.C.; Cavallucci, D.; Kadir, R.A.; Ng, K.W.; Sorooshian, S. TRIZ trend of engineering system evolution: A review on applications, benefits, challenges and enhancement with computer-aided aspects. *Comput. Ind. Eng.* **2022**, *174*, 108833. [[CrossRef](#)]
24. Ghane, M.; Ang, M.C.; Cavallucci, D.; Kadir, R.A.; Ng, K.W.; Sorooshian, S. Semantic TRIZ feasibility in technology development, innovation, and production: A systematic review. *Heliyon* **2024**, *10*, e23775. [[CrossRef](#)]
25. Jiang, S.; Li, W.; Qian, Y.; Zhang, Y.; Luo, J. AutoTRIZ: Automating engineering innovation with TRIZ and large language models. *Adv. Eng. Inform.* **2025**, *65*, 103312. [[CrossRef](#)]
26. Ni, X.; Samet, A.; Cavallucci, D. Similarity-based approach for inventive design solutions assistance. *J. Intell. Manuf.* **2022**, *33*, 1681–1698. [[CrossRef](#)]
27. Yan, W.; Liu, H.; Zanni-Merk, C.; Cavallucci, D. IngeniousTRIZ: An automatic ontology-based system for solving inventive problems. *Knowl.-Based Syst.* **2015**, *75*, 52–65. [[CrossRef](#)]
28. Anderson, J.R. ACT: A simple theory of complex cognition. *Am. Psychol.* **1996**, *51*, 355. [[CrossRef](#)]
29. Anderson, J.R.; Bothell, D.; Byrne, M.D.; Douglass, S.; Lebiere, C.; Qin, Y. An integrated theory of the mind. *Psychol. Rev.* **2004**, *111*, 1036. [[CrossRef](#)] [[PubMed](#)]
30. Laird, J.E.; Newell, A.; Rosenbloom, P.S. Soar: An architecture for general intelligence. *Artif. Intell.* **1987**, *33*, 1–64. [[CrossRef](#)]
31. Laird, J.E. *The Soar Cognitive Architecture*; MIT press: Cambridge, MA, USA, 2019.
32. Chong, H.-Q.; Tan, A.-H.; Ng, G.-W. Integrated cognitive architectures: A survey. *Artif. Intell. Rev.* **2007**, *28*, 103–130. [[CrossRef](#)]

33. Kotseruba, I.; Tsotsos, J.K. 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artif. Intell. Rev.* **2020**, *53*, 17–94. [\[CrossRef\]](#)
34. Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; Wen, J.-R. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–47. [\[CrossRef\]](#)
35. Töberg, J.-P.; Ngonga Ngomo, A.-C.; Beetz, M.; Cimiano, P. Commonsense knowledge in cognitive robotics: A systematic literature review. *Front. Robot. AI* **2024**, *11*, 1328934. [\[CrossRef\]](#)
36. Collins, K.M.; Sucholutsky, I.; Bhatt, U.; Chandra, K.; Wong, L.; Lee, M.; Zhang, C.E.; Zhi-Xuan, T.; Ho, M.; Mansinghka, V. Building machines that learn and think with people. *Nat. Hum. Behav.* **2024**, *8*, 1851–1863. [\[CrossRef\]](#)
37. Sukhobokov, A.; Belousov, E.; Gromozdov, D.; Zenger, A.; Popov, I. A universal knowledge model and cognitive architectures for prototyping AGI. *Cogn. Syst. Res.* **2024**, *88*, 101279. [\[CrossRef\]](#)
38. Summers, T.; Yao, S.; Narasimhan, K.; Griffiths, T. Cognitive architectures for language agents. *Trans. Mach. Learn. Res.* **2023**. [\[CrossRef\]](#)
39. Bisk, Y.; Holtzman, A.; Thomason, J.; Andreas, J.; Bengio, Y.; Chai, J.; Lapata, M.; Lazaridou, A.; May, J.; Nisnevich, A. Experience grounds language. *arXiv* **2020**, arXiv:2004.10151. [\[CrossRef\]](#)
40. Marcus, G. Deep learning: A critical appraisal. *arXiv* **2018**, arXiv:1801.00631. [\[CrossRef\]](#)
41. Xu, Q.; Peng, Y.; Nastase, S.A.; Chodorow, M.; Wu, M.; Li, P. Large language models without grounding recover non-sensorimotor but not sensorimotor features of human concepts. *Nat. Hum. Behav.* **2025**, *9*, 1871–1886. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [\[CrossRef\]](#)
43. Farkaš, I.; Vavrečka, M.; Wermter, S. Will multimodal large language models ever achieve deep understanding of the world? *Front. Syst. Neurosci.* **2025**, *19*, 1683133. [\[CrossRef\]](#)
44. Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; Chen, E. A survey on multimodal large language models. *Natl. Sci. Rev.* **2024**, *11*, nwae403. [\[CrossRef\]](#)
45. Peng, Z.; Wang, W.; Dong, L.; Hao, Y.; Huang, S.; Ma, S.; Wei, F. Kosmos-2: Grounding multimodal large language models to the world. *arXiv* **2023**, arXiv:2306.14824. [\[CrossRef\]](#)
46. Wheeler, D.; Tripp, E.E.; Natarajan, B. Semantic communication with conceptual spaces. *IEEE Commun. Lett.* **2022**, *27*, 532–535. [\[CrossRef\]](#)
47. Yu, D.; Yang, B.; Liu, D.; Wang, H.; Pan, S. A survey on neural-symbolic learning systems. *Neural Netw.* **2023**, *166*, 105–126. [\[CrossRef\]](#)
48. Gardenfors, P. *Conceptual Spaces: The Geometry of Thought*; MIT press: Cambridge, MA, USA, 2004.
49. d’Avila Garcez, A.S.; Lamb, L.C.; Gabbay, D.M. *Neural-Symbolic Cognitive Reasoning*; Springer: Berlin/Heidelberg, Germany, 2009.
50. Lamb, L.C.; Garcez, A.; Gori, M.; Prates, M.; Avelar, P.; Vardi, M. Graph neural networks meet neural-symbolic computing: A survey and perspective. *arXiv* **2020**, arXiv:2003.00330.
51. Garcez, A.d.A.; Lamb, L.C. Neurosymbolic ai: The 3 rd wave. *Artif. Intell. Rev.* **2023**, *56*, 12387–12406. [\[CrossRef\]](#)
52. Bhuyan, B.P.; Ramdane-Cherif, A.; Tomar, R.; Singh, T. Neuro-symbolic artificial intelligence: A survey. *Neural Comput. Appl.* **2024**, *36*, 12809–12844. [\[CrossRef\]](#)
53. Van Meter, H.J. Revising the DIKW pyramid and the real relationship between data, information, knowledge, and wisdom. *Law Technol. Hum.* **2020**, *2*, 69–80. [\[CrossRef\]](#)
54. Calzati, S. An ecosystemic view on information, data, and knowledge: Insights on agential AI and relational ethics. *AI Ethics* **2025**, *5*, 3763–3776. [\[CrossRef\]](#)
55. Duan, Y. Bridging the gap between purpose-driven frameworks and artificial intelligence. *Appl. Sci.* **2023**, *13*, 10747. [\[CrossRef\]](#)
56. Wu, X.; Zhang, W.; Wang, Y.; Zhang, H.; Bai, S. DIKWP Framework: Intelligentization, Green Innovation, and Servitization for Manufacturing Sustainability in Circular Economy. *Sage Open* **2026**, *16*, 21582440251415300. [\[CrossRef\]](#)
57. Zhan, S.; Tang, F.; Li, Y.; Mei, Q.; Lu, Z.; Li, X.-S.; Zhang, B. Construction of Index System of International Economic Cooperation Innovation Model in Digital Era-based on TIF Theory and DIKWP Model. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*; IEEE: New York, NY, USA, 2023; pp. 809–812.
58. Wei, H.; Kong, Q.; Wang, X.; He, H.; Wu, H.; Li, X. DIKWP construction of public hospital performance evaluation index system-based on TIF model domain wealth theory. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*; IEEE: New York, NY, USA, 2023; pp. 821–828.
59. Yao, L.; Liu, L.; Wei, X. A Study of Educational Curriculum Data Utilization Paths Based on the DIKWP Model. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*; IEEE: New York, NY, USA, 2023; pp. 1140–1147.

60. Liu, L.; Liu, R.; Liu, X. Application of DIKWP Model to Optimize the Allocation of Postgraduate Education Funds. In *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*; IEEE: New York, NY, USA, 2023; pp. 1153–1160.
61. Mei, Y.; Duan, Y. Bidirectional Semantic Communication Between Humans and Machines Based on Data, Information, Knowledge, Wisdom, and Purpose Artificial Consciousness. *Appl. Sci.* **2025**, *15*, 1103. [\[CrossRef\]](#)
62. Mei, Y.; Duan, Y. A Review of Personalized Semantic Secure Communications Based on the DIKWP Model. *Electronics* **2025**, *14*, 3671. [\[CrossRef\]](#)
63. Li, Y.; Duan, Y.; Maamar, Z.; Che, H.; Spulber, A.-B.; Fuentes, S. Swarm Differential Privacy for Purpose-Driven Data-Information-Knowledge-Wisdom Architecture. *Mob. Inf. Syst.* **2021**, *2021*, 6671628. [\[CrossRef\]](#)
64. Vu, S.; Duan, Y.; Pham, U.; Song, M.; Guo, Z.; Mei, Y.; Nguyen, H.D. The Role of DIKWP Assets in Shaping Economic Systems and Implications for an Intelligent Consultant System in Real Estate Investment. *J. Cases Inf. Technol. (JCIT)* **2024**, *26*, 1–27. [\[CrossRef\]](#)
65. Wu, K.; Duan, Y. Modeling and Resolving Uncertainty in DIKWP Model. *Appl. Sci.* **2024**, *14*, 4776. [\[CrossRef\]](#)
66. Russo, D.; Spreafico, C. TRIZ 40 Inventive principles classification through FBS ontology. *Procedia Eng.* **2015**, *131*, 737–746. [\[CrossRef\]](#)
67. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11809–11822.
68. Creswell, A.; Shanahan, M.; Higgins, I. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv* **2022**, arXiv:2205.09712. [\[CrossRef\]](#)
69. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
70. Min, S.; Michael, J.; Hajishirzi, H.; Zettlemoyer, L. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 5783–5797.
71. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 38154–38180.
72. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language models can teach themselves to use tools. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 68539–68551.
73. Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Roziere, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A. Augmented Language Models: A Survey. *arXiv* **2023**, arXiv:2302.07842. [\[CrossRef\]](#)
74. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
75. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large language models are zero-shot reasoners. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 22199–22213.
76. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 157–173. [\[CrossRef\]](#)
77. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual Acm Symposium on User Interface Software and Technology*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–22.
78. Kuhn, L.; Gal, Y.; Farquhar, S. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv* **2022**, arXiv:2212.07769.
79. Mündler, N.; He, J.; Jenko, S.; Vechev, M. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv* **2023**, arXiv:2305.15852.
80. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27730–27744.
81. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C. Constitutional ai: Harmlessness from ai feedback. *arXiv* **2022**, arXiv:2212.08073. [\[CrossRef\]](#)
82. Luo, H.; Specia, L. From understanding to utilization: A survey on explainability for large language models. *arXiv* **2024**, arXiv:2401.12874. [\[CrossRef\]](#)
83. Turpin, M.; Michael, J.; Perez, E.; Bowman, S. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 74952–74965.
84. Mökander, J.; Schuett, J.; Kirk, H.R.; Floridi, L. Auditing large language models: A three-layered approach. *AI Ethics* **2024**, *4*, 1085–1115. [\[CrossRef\]](#)

85. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv* **2023**, arXiv:2303.12712. [[CrossRef](#)]
86. Guo, T.; Yang, Q.; Wang, C.; Liu, Y.; Li, P.; Tang, J.; Li, D.; Wen, Y. Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. *Complex Intell. Syst.* **2024**, *10*, 7063–7076. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.