Inventive Problem Solving with LLMs: A Benchmark for TRIZ Reasoning

Anonymous ACL submission

Abstract

Large language models have been widely used in invention workflows, but effective support requires more than open-ended generative ideation. TRIZ offers a structured framework that can guide LLMs in inventive problem reasoning. However, evaluations in prior work are small-scale and rarely grounded in patent text. We introduce TRIZBENCH, a dataset and benchmark for TRIZ reasoning grounded in open technical sources and U.S. patents. Furthermore, we design three tasks covering core TRIZ workflow stages, including contradiction prediction, inventive principle prediction, and grounded TRIZ reasoning. Experiments with multiple LLM baselines show that detecting contradictions is easier than recovering correct trade-off pairs, and principle prediction benefits from TRIZ structured reasoning. Our findings also underscore the importance of grounding: semantic retrieval enables evidence-based justifications and helps explain why LLMs fail. Dataset and codes are available here: https://anonymous.4open.science/r/trizbench-E519.

1 Introduction

Large language models (LLMs) are increasingly used as assistants in invention workflows (Ma et al., 2023; Guo et al., 2025a), supporting tasks such as summarizing prior art (Sharma et al., 2019; Wang et al., 2024b), reframing problem statements (Einarsson et al., 2024), and proposing ideas or solutions (Noy and Zhang, 2023; Hou et al., 2024; Chen et al., 2024). However, effective invention support requires more than generating plausible text (Siddharth and Luo, 2024). It depends on correctly identifying the underlying trade-off in a technical system and producing reasoning that can be referred to the problem–solution narratives.

TRIZ (Theory of Inventive Problem Solving) (Altshuller, 1999) provides a structured reasoning framework for this setting: it represents problems as *contradictions* and organizes common resolution patterns as *inventive principles*. For instance, US9683836 (Sandhawalia et al., 2017) involves a trade-off between deploying laser scanners in sensitive locations and limitations in efficiency. The resolution then is to enrich the feature representation used by the classifier, which aligns with TRIZ principle 35 (Parameter Changes) (Li et al., 2022).

Patents are a natural corpus for TRIZ reasoning because they contain problem–solution narratives at scale (Wang et al., 2016; Chang et al., 2017). However, patent language makes TRIZ structure difficult to recover automatically. For example, trade-offs in patent text are often implicit, dispersed across sections (e.g., background, limitations, claims), and expressed in domain-specific phrasing that does not align well with TRIZ descriptions (Guarino et al., 2020, 2022; Ali et al., 2024). As a result, it remains unclear whether LLMs can reliably apply TRIZ reasoning to patents while providing accurate predictions with text-based justifications.

Recent TRIZ related LLMs work, such as AutoTRIZ (Jiang and Luo, 2024) and TRIZ-GPT (Chen et al., 2024), demonstrate promising pipelines for contradiction identification and principle-guided ideation. However, their evaluations largely rely on small case collections and domain-specific design scenarios (Xie and Liu, 2023; Guo et al., 2025b). More broadly, the field lacks a large-scale benchmark that (i) covers the key steps of the TRIZ workflow, (ii) supports evaluation under patent domain shift, and (iii) enables evidence-based verification of model reasoning.

In this work, we introduce **TRIZBENCH**, a dataset and benchmark for TRIZ reasoning grounded in technical/research papers and patents. TRIZBENCH includes **1,354** TRIZ cases collected from domain-expert sources across diverse technical areas. Each case is represented with a structured schema covering system context, improve–worsen

trade-offs, solutions and principles, and supporting evidence spans. To evaluate transfer to real invention documents, we additionally provide a patent benchmark of **429** US patents, including a subset with *human-labeled* TRIZ parameters and principles. This case–patent design supports learning both contradiction structure and principle selection from cases, then transferring to patent text, where labels are sparse and trade-offs are rarely expressed explicitly.

We define three benchmark tasks that capture core stages of the TRIZ workflow and progressively increase difficulty. **Task 1 (Contradiction prediction)** evaluates whether models can recover the improve–worsen pair from technical descriptions. **Task 2 (Inventive principle prediction)** evaluates inventor-associated principle prediction for a given contradiction and quantifies the extent to which methods exploit TRIZ structure. **Task 3 (Grounded TRIZ reasoning)** requires sentence-level evidence from patent text to justify predicted parameters/principles, enabling grounded outputs for downstream patent workflows.

Our Contributions. We provide:

- **TRIZBENCH:** a large, source-grounded corpus of 1,354 TRIZ cases and 429 patents, including a human-labeled patent subset with TRIZ parameters and principles.
- **Benchmarks:** three tasks spanning contradiction pair prediction, inventor-associated principle prediction, and grounded TRIZ reasoning with sentence-level evidence.
- **Findings:** a comprehensive comparison of prompting, fine-tuning, and retrieval baselines across multiple models, highlighting failure modes under patent domain shift and the value of grounding-based evaluation beyond accuracy.

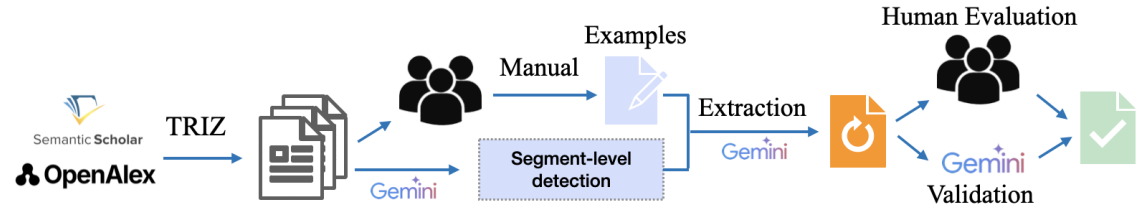
2 Related work

LLMs for ideation and invention. Recent work has demonstrated LLMs as general-purpose assistants for ideation and invention across many domains, including creative concept exploration in specialized domains (Hou et al., 2024), language-driven generation of design concepts (Zhu and Luo, 2022; Filippi, 2023), and grounded scientific ideation from research papers (Radensky et al., 2024). Moreover, patent-focused LLMs aim to support IP workflows by generating or structuring patent content and concepts, which shows growing interest in applying LLMs to legal and technical documents (Ren et al., 2025; Bai et al., 2024; Wang et al., 2024a; Guo et al., 2025a). However, open-ended ideas are hard to validate without domain experts. Thus, it remains unclear how LLMs generate ideas and solve problems. TRIZ-inspired approaches address this by using contradictions and guiding principle-based ideation, including end-to-end TRIZ pipelines and interactive TRIZ assistants, as well as TRIZ-augmented ideation tools with LLMs (Jiang and Luo, 2024; Chen et al., 2024; Lee et al., 2024; Guo et al., 2025b). Despite these advances, evaluation remains largely system-driven and case-study based. Existing TRIZ-LLM papers typically report qualitative analyses or small case sets in specific domains rather than evaluating on a large-scale benchmark with evidence supervision for measuring contradiction detection, principle attribution, and patent-domain grounding.

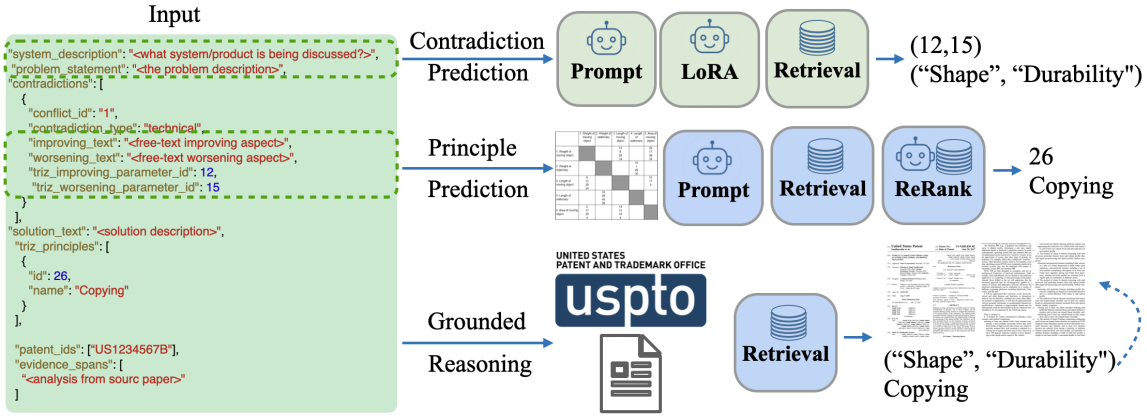
TRIZ for patent mining. Prior work has explored extracting TRIZ structure from patent text. Early systems typically combined rule-based or pattern-driven processing with TRIZ resources to locate contradictions and principles in patents (Cascini and Russo, 2007; Souili and Cavallucci, 2012; Souili et al., 2015; Wang et al., 2016; Chang et al., 2017). More recent work introduces deep learning methods that explicitly separate where the contradiction is stated from patent sentences (Guarino et al., 2020, 2022; Trapp and Warschat, 2024). Furthermore, patent-focused TRIZ applications have also explored evolution trends to identify promising patents for technology transfer (Park et al., 2013; Yun et al., 2022). However, evaluation is typically conducted on individual components in TRIZ in these works. Thus, it does not yield a comprehensive benchmark that jointly verify contradiction detection, inventor principle prediction, and sentence-level grounding on patent language with LLMs.

3 TRIZBENCH Dataset

We introduce **TRIZBENCH**, a dataset and benchmark for TRIZ reasoning and inventive problem solving. TRIZBENCH consists of structured cases from open scholarly and technical sources. Each case records (i) system context, (ii) a contradiction framed as an improve–worsen trade-off pair,



(a) TRIZBENCH construction pipeline



(b) Overview of benchmark tasks with TRIZBENCH

Figure 1: Overall framework of TRIZBENCH. The construction pipeline of TRIZBENCH are three stages, including seed human annotation, automatic extraction with LLMs and human evaluation. The benchmark tasks include three tasks on case and patent corpus separately. We evaluate multiple LLM baselines across all tasks.

(iii) a solution mechanism and associated inventive principles, and (iv) supporting evidence spans that ground these fields in the source text.

Design goals. TRIZBENCH is designed to: (1) support evidence-supported contradiction identification from natural language; (2) provide paired contradiction–resolution representations for downstream retrieval and generation; (3) enable both prompting-based evaluation and supervised learning from structured annotations; and (4) connect case-based TRIZ reasoning to patent-focused applications, including interpretable patent analytics and drafting assistance.

3.1 Data collection and preprocessing

Data sources. We collect TRIZ-relevant documents from two open-access sources: (1) peer-reviewed papers available as public PDFs and (2) TRIZ-focused web publications (HTML), including The TRIZ Journal[1] and TRIZ community proceedings[2]. For papers, we query open indexing services (e.g., Semantic Scholar and OpenAlex) with TRIZ-related keywords (e.g., *TRIZ*, *technical/-physical contradiction*, *inventive principles*, *ARIZ*) from 2000 to 2025.

Preprocessing. We retain documents that are English, contain substantive technical content which are beyond introductions and surveys, and are parsable by our pipeline. We remove duplicates and non-technical pages. Then, we convert PDFs into structured markdown using `marker`[3] with `Gemini-2.5-flash` (Comanici et al., 2025) as the backend, preserving layout structure especially for tables/figures. We apply lightweight normalization and maintain approximate source-location pointers for evidence checking. Finally, we segment each document into coherent units (sections/subsections when available; otherwise paragraphs) to reduce context length and improve extraction stability. HTML sources are converted to the same markdown format and processed with identical normalization and segmentation.

3.2 TRIZ Case Extraction Pipeline

We extract benchmark TRIZ cases from heterogeneous technical documents using a three-stage pipeline: (i) define a fixed, evidence-grounded schema via human-annotated seed annotation; (ii) classify segments with a lightweight case detector;

[1] https://the-trizjournal.com
[2] https://trizfest.org
[3] https://github.com/datalab-to/marker

and (iii) perform structured extraction with automatic validation and normalization.

Stage I: Seed annotation. We (two human experts) manually annotate a small seed set, including 10 documents, 64 segments, and 17 cases, to operationalize what constitutes a TRIZ case in natural technical writing and to refine a fixed output schema. A segment qualifies as a case if it describes a concrete system, a contradiction (technical or physical), and an explicit or implied resolution. We normalize all cases into an improve–worsen pair (`improving_text`, `worsening_text`) and require evidence spans for the contradiction and the resolution to support downstream benchmarking.

Stage II: Segment-level detection. To reduce cost and false positives, we first predict whether each segment contains at least one extractable case. We implement this step by prompting `Gemini-2.5-pro` with seed exemplars to produce a binary label (`case` vs. `no_case`). Only `case` segments proceed to structured extraction. `no_case` segments are retained for analyzing false positives in case detection.

Stage III: Structured extraction. For each candidate segment, we prompt `Gemini-2.5-pro` to output one or more case objects in the defined schema in Stage I. The extractor is instructed to (i) fill required fields, (ii) adhere to the source text, (iii) attach evidence spans for major fields, and (iv) abstain when key components are missing. When explicitly stated, we also extract auxiliary metadata such as referenced patent IDs and TRIZ parameter/principle identifiers.

Validation and normalization. We automatically validate all extracted objects with schema checks and evidence-grounding checks. Invalid outputs are discarded; minor omissions are filled with `null` defaults. We further normalize vocabularies (e.g., contradiction types) and deduplicate near-duplicate cases within a document.

Outputs. The pipeline produces three outputs, including a validated case corpus, a segment manifest with detection labels for reproducibility, and logs summarizing validation failures.

3.3 Patent Corpus Construction

We augment TRIZBENCH with a patent corpus to evaluate TRIZ reasoning under domain shift. We include two sources of patents: (i) patents explicitly referenced in extracted TRIZ cases, and (ii) an auxiliary set of 234 U.S. patents with TRIZ annotations referring to TRIZ-focused sources. For each patent, we normalize the patent identifier and query PatentsView to retrieve a set of fields, such as title, abstract, background/summary when available, claims, CPC codes, grant date. We restrict the release to U.S. patents in the current version.

This paired case–patent design supports evaluation and transfer: models can learn contradiction structure and principle usage from case narratives and then apply it to patent language, where supervision is sparse and trade-offs are often stated indirectly.

3.4 Human Evaluation

Automatic validation (Stage III) enforces schema and evidence constraints at scale, but it cannot fully assess semantic correctness. Therefore, we conduct human evaluation to estimate the quality of both extracted TRIZ cases and patent corpus. Two human experts with experience reading technical papers and patents annotated the extracted cases and patents following a detailed guideline. Additionally, they conducted two rounds of discussion to resolve disagreements and reached consensus decisions (approve/reject).

Case quality. We sample **(N=200)** extracted cases and evaluate: (1) trustworthiness as whether each extracted field is supported by the cited evidence; (2) contradiction correctness as whether `improving_text` and `worsening_text` accurately reflect the trade-off described in the source; (3) completeness as whether required fields are present and non-empty; and (4) resolution correctness as whether `solution_text` captures the stated mechanism without introducing unsupported claims. Each case is assigned one of three outcomes: `approve`, `edit`, or `reject`. Overall, nearly all cases are approved as-is. The few interventions are minor edits aimed at improving clarity. We evaluate the validity of patent linkage separately below, since patent identifiers may be mentioned without constituting a patent-centered analysis.

Patent quality. Because patents can be mentioned without substantive analysis, we also evaluate the cases with non-empty `patent_ids`. Annotators verify whether the cited patents are supported by the source evidence, and whether the patent is used to analyze the contradiction or the solution

TRIZ case corpus		Patent corpus	
Statistic	**Number**	**Statistic**	**Number**
Source documents	590	Patent-linked cases	148 (10.9%)
Cases	1,354	Unique case-linked patents	223
Contradictions (total)	1,679	Auxiliary labeled patents	234
Solutions (total)	2,630	Unique patents (total)	429
Contradictions w/ Triz parameters†	491 (29.2%)	Patents w/ parameters	257 (56.2%)
Cases w/ TRIZ principles	734 (54.2%)	Patent w/ principles	304 (66.5%)
Major domains	10	CPC codes (3-digit)	83
Case text length (tokens)‡	278 (avg.)	Patents per patent-linked case	1.96 (avg.)
Contradiction evidence (tokens)	53 (avg.)	Patent abstract (tokens)	187 (avg.)
Principle evidence (tokens)	41 (avg.)	Patent first claims (tokens)	216 (avg.)

Table 1: Overall dataset statistics for **TRIZBENCH**. †Counts contradictions where both improving and worsening TRIZ parameter IDs are present. ‡Case text length is computed over {`system_description`, `problem_statement`, `solution_text`} when present.

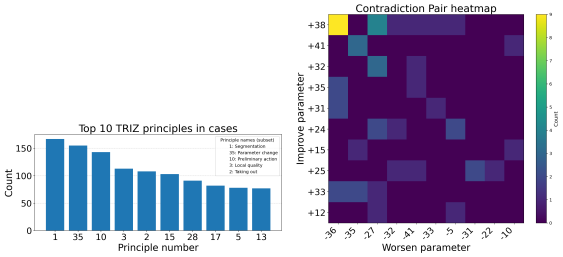
rather than a toy example. We additionally record error categories such as false links and ambiguous patent mentions. In total, we review 148 patent-link cases which contain non-empty `patent_ids`. Of these, 4/148 are flagged as non-substantive, which means patents mentioned but not used as a patent case and then excluded from patent benchmarks.

3.5 Dataset statistics

Table 1 summarizes **TRIZBENCH**: 1,354 cases with 1,679 contradiction entries; 491 contradictions (29.2%) include improving–worsening TRIZ parameter pairs and 734 cases (54.2%) include TRIZ principles (2,630 total possible solutions/principles). Figure 2 shows that the most principles mentioned in cases are "Segmentation", and the contradiction in patent is ("Extend of automation", "Device complexity"). The case narrative length averages 278 tokens, while evidence spans are shorter. We map fine-grained domain tags into 10 major domains , including Management, Mechanical/-Manufacturing, Computer/Electronics, Biomedical/Pharma and so on (Appendix A.2.1). The patent corpus contains 429 unique patents which span 83 CPC 3-digit classes, such as A61 (Medical), G06 (Computing) and H01 (Electric). Detailed statistics information of patents are in Appendix A.2.2.

4 Experiments

We evaluate models on three tasks that probe TRIZ inventive reasoning from both case narratives and patent language: (1) *Contradiction prediction*, which extracts an improve–worsen trade-off and (when available) maps it to classical TRIZ parameters; (2) *Inventive principle prediction*, whether models can recover the inventors TRIZ principles used in the source case or patent; and (3) *Grounded TRIZ reasoning*, which additionally requires sentence-level evidence citations from patent text to justify predicted principles and mechanisms. In all experiment settings, we use an 80/20 train/-validation split for both the case corpus and the patent corpus, with splits performed at the case/-patent level. All open-source model (Qwen3 (Yang et al., 2025) and LLaMA3 (Dubey et al., 2024)) experiments are run on a cluster with 4×A100 GPUs, while Gemini (Comanici et al., 2025) results are obtained via the Google Gemini API under the same task prompts and evaluation protocols. More experiments details are in Appendix.



(a) Top 10 principles in cases (b) Top 10 pairs in patents

Figure 2: Top Contradiction pairs and principles distributions in TRIZBENCH.

4.1 Contradiction prediction

Goal. This task evaluates whether a model can predict the two aspects of a TRIZ contradiction from technical descriptions: the *improving* (t^+/p^+; what we want to improve) and the *worsening* (t^-/p^-; what deteriorates as a trade-off). We emphasize *pair-level* correctness because further TRIZ reasoning, such as retrieving inventive principles or generating a resolution, depends on jointly identifying the improving and worsening aspects.

Model	Method	Case		Patent
		HasF1	PairF1	Hit@3
Qwen3-7B	ZS	0.84	0.31	0.13
Qwen3-7B	FS	0.87	0.30	0.15
Qwen3-32B	ZS	0.76	0.29	0.21
Qwen3-32B	FS	0.75	0.34	0.19
LLaMA3.1-8B	ZS	0.77	0.29	0.14
LLaMA3.1-8B	FS	0.83	0.32	0.14
LLaMA3.1-70B	ZS	0.80	0.33	0.16
LLaMA3.1-70B	FS	0.76	0.36	0.19
Gemini-2.5-Pro	ZS	0.77	0.42	0.24
Gemini-2.5-Pro	FS	0.82	**0.45**	0.26
PatentSBERTa	Retrieval	–	–	0.23
BGE	Retrieval	–	–	0.17
Qwen3-7B	LoRA	**0.90**	0.43	**0.28**
LLaMA3.1-8B	LoRA	0.88	0.38	0.26

Table 2: Main results on contradiction prediction. **HasF1** is F1 for contradiction detection (case-level). **PairF1** is semantic matching F1 for improve–worsen contradiction pairs at threshold $\tau = 0.65$ (case-level). **Hit@3** reports whether a gold pair appears among the model's top-3 ranked predicted pairs (patent-level).

We evaluate (1) extraction of the improving/worsening *text* and (2) when labels are available, prediction of the corresponding TRIZ *parameter IDs* in the classical 39-parameter space.

4.1.1 Case-level prediction

Data. Each case provides `system_description` and `problem_statement`, and includes one or more annotated contradictions with gold `improving_text` and `worsening_text`. **Input.** The input x concatenates the case narrative: $x =$ [`system_description`; `problem_statement`], **Output.** Models output a set of candidate pairs $(\hat{t}^+, \hat{t}^-)$.

4.1.2 Patent-level prediction

Data. We construct a parallel benchmark over patents. Each patent is annotated with one improving parameter ID and one worsening parameter ID. **Input.** For each patent, we build x by concatenating text from available sections, including *abstract*, *background*, *summary*, and the *first independent claim*. **Output.** Models predict $(\hat{p}^+, \hat{p}^-)$ in the 39-parameter ID space. This setting probes domain transfer: patent documents are claim-centered and more formal, but trade-offs often stated implicitly.

4.1.3 Methods

We benchmark three approach families. (1) **LLM prompting:** zero-shot (ZS) and few-shot (FS), where FS prepends k labeled exemplars sampled randomly from the train set. Prompts enforce structured outputs and instruct models to abstain when a contradiction cannot be grounded. (2) **LoRA fine-tuning:** LoRA adapters trained under the same output schema on the case and patent train splits. We evaluate case-LoRA, patent-LoRA, and transfer (case-LoRA applied to patents) to measure domain shift. (3) **Retrieval baselines:** for patent-level parameter prediction, we rank TRIZ parameters by embedding similarity between the patent text and each parameter's name and description using sentence encoders (Bekamiri et al., 2024; Chen et al., 2025).

4.1.4 Evaluation

We evaluate contradiction detection and improve–worsen pair quality, with pair-level correctness as the primary metric.

Contradiction detection. We predict whether an input contains a contradiction and report F1, capturing abstention on non-contradiction inputs and avoiding hallucinated structures.

Improve–worsen pair matching (case). We align predicted $(\hat{t}^+, \hat{t}^-)$ to gold (t^+, t^-) using embedding-based semantic matching with `Qwen3-Embedding-8B`. Pair similarity is the average of cosine similarities on the improving and worsening aspects. Note that, we allow swapping and take the maximum under swapped alignment. We perform one-to-one matching under threshold τ to compute pair-level precision/recall/F1, and report F1 at τ=0.65 (additional results in A.3.2).

Parameter prediction (patent). Models output TRIZ parameter IDs $(\hat{p}^+, \hat{p}^-)$ in the 39-parameter space. We report PairHit@K requiring both aspects correct within top-K (K=3) (additional results are in Appendix A.3.2).

Results. Table 2 shows contradiction detection is easier than recovering the correct improve–worsen pair. Gemini-2.5-Pro achieves the best ZS/FS PairF1 on cases (0.42–0.45) and competitive Hit@3 on patents (0.24–0.26), while LoRA further improves detection and pair retrieval (best detection F1: 0.90; best patent Hit@3: 0.28), suggesting structured learning helps under semantic evaluation. Retrieval is also competitive on patents (PatentSBERTa Hit@3=0.23), indicating that mapping patent phrasing to contradiction aspects is a

key challenge. Few-shot gains are small and sometimes inconsistent, likely due to exemplar sensitivity. We highlight directions including dependency-aware pair extraction, hybrid retrieve and loRA pipelines for domain transfer to patent, and end-to-end grounded training to downstream principle reasoning (Sections 4.2, 4.3).

4.2 Inventive principle prediction

Goal. This task evaluates whether a model can predict the *inventive principles used in the source* to address a contradiction. Given a technical context and a single contradiction with an improve–worsen pair, the model outputs a ranked list of TRIZ principles (IDs 1–40) in the classical set. Because sources may cite multiple principles for the same contradiction, we treat the task as ranked multi-label prediction.

4.2.1 Case principle prediction

Data. We include a TRIZ case when it contains at least one contradiction and at least one labeled inventive principle. Since many cases include multiple contradictions, we use a single-contradiction protocol by selecting one contradiction per case, yielding one instance $(x, \mathcal{G})$, where $\mathcal{G}$ is the gold principle set. **Input.** x follows a structured template concatenating `system_description`, `problem_statement`, and the selected contradiction `improve_text`/`worsen_text`. **Output.** Models return a ranked list $\hat{\mathbf{z}}$ of principle IDs in $[1, 40]$, truncated to top-K ($K = 1, 3, 5$).

4.2.2 Patent principle prediction

Data. We construct an similar benchmark over patents using our classical TRIZ labeled patent set: given patent text x, models output a ranked list of principle IDs in the same 40-principle space.

4.2.3 Methods

We benchmark five approach families: (1) **classical matrix lookup** from the TRIZ contradiction matrix; (2) **text retrieval** ranking principles by similarity to principle names/descriptions; (3) **LLM prompting** in zero-shot (ZS) and few-shot (FS) settings (FS prepends in-context examples); (4) **retrieval reranking** that reranks matrix candidates with an LLM; and (5) **end-to-end pipelines** that first perform contradiction prediction (Section 4.1) and then aggregate matrix recommendations over predicted (p^+, p^-) pairs to produce a reranked principle list.

Model	Method	Case		Patent	
		Hit@3	Recall@3	Hit@3	Recall@3
Non-LLM baselines					
–	Matrix	0.67	0.26	0.48	0.39
–	TF-IDF	0.69	0.35	0.56	0.42
LLM baselines					
Qwen3-7B	ZS	0.39	0.12	0.27	0.18
Qwen3-7B	FS	0.51	0.19	0.25	0.15
Gemini-2.5-Pro	ZS	0.45	0.23	0.31	0.26
Gemini-2.5-Pro	FS	0.52	0.21	0.35	0.27
Retrieval reranking					
Qwen3-7B	M-Rerank	0.71	0.29	0.59	0.47
LLaMA3.1-8B	M-Rerank	0.68	0.24	0.55	0.36
End-to-end pipelines					
Qwen3-7B	ZS-Rerank	0.59	0.17	0.46	0.40
Qwen3-7B	LoRA-Rerank	0.65	0.26	0.52	0.44

Table 3: Main results on inventive principle prediction. We report Hit@3 and Recall@3 for both case and patent benchmarks. M-rerank denotes Matrix-Rerank which prompts an LLM to rerank matrix candidates.

4.2.4 Evaluation

Metrics. Since gold labels are multi-principle, we report ranking metrics: **Hit@K**, whether any gold principle appears in the top-K, and **Recall@K**, the fraction of gold principles recovered in the top-K.

Results. Table 3 shows that principle prediction is largely driven by whether a method exploits TRIZ structure. Matrix lookup performs well because, given an improve–worsen parameter pair, the contradiction matrix narrows candidates to a small set prescribed by TRIZ practice (Hit@3: 0.67 on cases; 0.48 on patents). TF-IDF retrieval is also competitive, suggesting that many principles have distinctive names/descriptions and lexical overlap is often sufficient under Hit@3.

Direct LLM prompting underperforms across domains, indicating that unconstrained generation often yields plausible but weakly anchored principles rather than the *source-associated* set. In contrast, Matrix-Rerank achieves the best overall results (Hit@3 up to 0.71 case / 0.59 patent), suggesting LLMs are most effective as selectors over matrix-consistent candidates. End-to-end pipelines further show that principle accuracy improves with stronger upstream parameter predictions (Section 4.1), motivating better pair consistency and grounded constraints (Section 4.3), especially when patent principle labels are sparse.

4.3 Grounded TRIZ reasoning

Goal. Grounded TRIZ reasoning (GTR) evaluates TRIZ contradiction and principle reasoning *grounded in patent text*. Because patents rarely

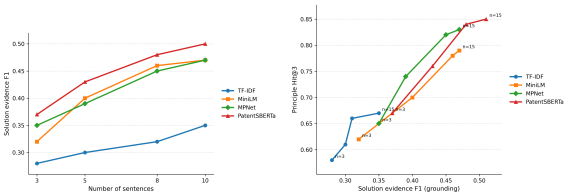
provide explicit TRIZ principle labels and often express trade-offs implicitly, we use richly annotated TRIZ cases as supervision and require sentence-level attributions (evidence citations to patent sentences) so that predicted parameters/principles are auditable and usable for downstream patent workflows, such as drafting problem–solution narratives and interpretable analytics.

Data. Each datapoint links a TRIZ case to one referenced US patent, since these cases include TRIZ specific patent analysis. We retrieve patent text via PatentsView, split it into sentences $\{s_i\}_{i=1}^{M}$, and provide the case contradiction as free text: improving aspect a^{imp} and worsening aspect a^{wor}. Models predict: (1) TRIZ parameters $(p^{\text{imp}}, p^{\text{wor}})$ (IDs or normalized names), (2) a ranked list of principle IDs $\pi_{1:K}$ (1–40), and (3) evidence sentence indices $E^{\text{imp}}, E^{\text{wor}}, E^{\text{sol}} \subset \{1, \ldots, M\}$ supporting the improving aspect, worsening aspect, and solution mechanism, respectively. We construct evidence sets via retrieval over patent sentences using anchors from case fields (details in Appendix A.3.4).

4.3.1 Methods and evaluation

We include (1) a **retrieval baseline** that retrieves evidence using anchors and outputs case-provided TRIZ labels, and (2) a **grounded attribution baseline** that maps aspects to parameters via nearest-neighbor matching to a parameter description, ranks candidate principles by matching principle descriptions to patent sentences, and outputs evidence sentence indices for each component. Embedding models include MiniLM (Wang et al., 2020), MPNet (Song et al., 2020), and PatentSBERTa (Bekamiri et al., 2024). We evaluate with evidence F1 against retrieved evidence sets and principle Hit@3 conditioned on whether the parameter pair is correct.

Results. In this task, we evaluate whether a model can justify predicted principles by citing patent sentences that describe the underlying mechanism. Figure 3a shows that evidence quality improves as the number of sentence increases, but depends strongly on the retriever: TF-IDF yields only modest gains, suggesting lexical matching is often insufficient for locating mechanism sentences, while grounded attribution baselines are consistently stronger and PatentSBERTa achieves the best alignment improving 0.14 at F1. Figure 3b shows a positive relationship between solution grounding



(a) Number of sentences (b) Grounding vs Accuracy

Figure 3: Grounding and its link to principle prediction. (a) Solution evidence F1 improves with more evidence sentences. (b) Improved grounding is associated with higher Principle Hit@3, indicating that evidence-supported solution mechanism retrieval contributes to more reliable TRIZ principle prediction.

(evidence F1) and principle prediction (Hit@3), indicating that reliable principle selection relies on retrieving semantically relevant mechanism descriptions. Moreover, qualitative examples show only labeled TRIZ accuracy can be misleading. For example, the model predicts the parameter pair right and Hit@3=1 yet misses evidence for the improving aspect in an athletic-glove case, while a vibrating-alarm timepiece case perfectly grounds the improvement mechanism (F1=1.0) despite an incorrect parameter pair. These results motivate GTR as a robust evaluation that tests whether predicted principles are actually supported by patent mechanisms.

5 Conclusion and Future Work

We introduced TRIZBENCH, a dataset and benchmark for TRIZ reasoning grounded in research papers and U.S. patents, with three tasks including trade-off contradiction prediction, inventive principle prediction, and grounded TRIZ reasoning with sentence-level evidence. Across multiple LLMs baselines, we find that it is difficult to extract correct trade-off pairs from patent text. We also observe that principle prediction is most reliable when methods exploit TRIZ structure, and grounding quality depends critically on semantic retrieval over patent sentences. Future work includes end-to-end training objectives that combine contradiction prediction with downstream principle selection, and integrating grounding as a important supervision signal for robust and trusty prediction. Indeed, these directions can further support downstream patent workflows such as claim drafting/rewriting and interpretable trend analysis.

Limitations

Our benchmark has several limitations. First, the TRIZ case corpus is extracted from open technical sources and is therefore subject to coverage and selection bias like domains, so extraction errors may persist despite validation and expert review. Moreover, the patent corpus is currently limited in size and scope to only U.S. patents. Since the available TRIZ labels are sparse, it may constrain how broadly we can assess principle prediction on patents. In addition, we focus on the classical TRIZ parameter/principle due to copyright issues of updated version of TRIZ matrix.

References

Amna Ali, Ali Tufail, Liyanage Chandratilak De Silva, and Pg Emeroylariffion Abas. 2024. Innovating patent retrieval: a comprehensive review of techniques, trends, and challenges in prior art searches. *Applied System Innovation*, 7(5):91.

Genrikh Saulovich Altshuller. 1999. *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical innovation center, Inc.

Zilong Bai, Ruiji Zhang, Linqing Chen, Qijun Cai, Yuan Zhong, Cong Wang, Yan Fang, Jie Fang, Jing Sun, Weikuan Wang, and 1 others. 2024. Patentgpt: A large language model for intellectual property. *arXiv preprint arXiv:2404.18255*.

Hamid Bekamiri, Daniel S Hain, and Roman Jurowetzki. 2024. Patentsberta: A deep nlp based hybrid model for patent distance and classification using augmented sbert. *Technological Forecasting and Social Change*, 206:123536.

Gaetano Cascini and Davide Russo. 2007. Computer-aided analysis of patents and search for triz contradictions. *International Journal of Product Development*, 4(1-2):52–67.

Hsiang-Tang Chang, Chen-Yen Chang, and Wen-Kuei Wu. 2017. Computerized innovation inspired by existing patents. In *2017 international conference on applied system innovation (ICASI)*, pages 1134–1137. IEEE.

Jianlyu Chen, Junwei Lan, Chaofan Li, Defu Lian, and Zheng Liu. 2025. Reasonembed: Enhanced text embeddings for reasoning-intensive document retrieval. *arXiv preprint arXiv:2510.08252*.

Liuqing Chen, Yaxuan Song, Shixian Ding, Lingyun Sun, Peter Childs, and Haoyu Zuo. 2024. Triz-gpt: An llm-augmented method for problem-solving. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88407, page V006T06A010. American Society of Mechanical Engineers.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hafsteinn Einarsson, Sigrún Helga Lund, and Anna Helga Jónsdóttir. 2024. Application of chatgpt for automated problem reframing across academic domains. *Computers and Education: Artificial Intelligence*, 6:100194.

Stefano Filippi. 2023. Measuring the impact of chatgpt on fostering concept generation in innovative product design. *Electronics*, 12(16):3535.

Guillaume Guarino, Ahmed Samet, and Denis Cavallucci. 2022. Patriz: A framework for mining triz contradictions in patents. *Expert Systems with Applications*, 207:117942.

Guillaume Guarino, Ahmed Samet, Amir Nafi, and Denis Cavallucci. 2020. Summatriz: summarization networks for mining patent contradiction. In *2020 19th IEEE international conference on machine learning and applications (ICMLA)*, pages 979–986. IEEE.

Xingyu Guo, Yi Tan, and Rui Chen. 2025a. Leveraging large language models and triz: A multi-agent framework for automated patent drafting and innovation generation. In *International TRIZ and Artificial Intelligence Conference*, pages 134–151. Springer.

Zishun Guo, Meng Song, Xiaofen Fang, Cuiyun Lin, Hengjie Zhang, Xiaoye Li, and Wenxiao Wang. 2025b. Exploring synergies between aigc and triz in the optimisation of road cone design through integrated innovation methods. *Journal of Engineering Design*, 36(2):256–275.

Yihan Hou, Manling Yang, Hao Cui, Lei Wang, Jie Xu, and Wei Zeng. 2024. C2ideas: Supporting creative interior color design ideation with a large language model. In *Proceedings of the 2024 CHI conference on human factors in computing systems*, pages 1–18.

Shuo Jiang and Jianxi Luo. 2024. Autotriz: Artificial ideation with triz and large language models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88377, page V03BT03A055. American Society of Mechanical Engineers.

CKM Lee, Jingying Liang, Kai Leung Yung, and Kin Lok Keung. 2024. Generating triz-inspired guidelines for eco-design using generative artificial intelligence. *Advanced Engineering Informatics*, 62:102846.

SP Li, KM Yu, YC Yeung, and KL Keung. 2022. Patent review and novel design of vehicle classification system with triz. *World Patent Information*, 71:102155.

Kevin Ma, Daniele Grandi, Christopher McComb, and Kosa Goucher-Lambert. 2023. Conceptual design generation using large language models. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87349, page V006T06A021. American Society of Mechanical Engineers.

Shakked Noy and Whitney Zhang. 2023. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192.

Hyunseok Park, Jason Jihoon Ree, and Kwangsoo Kim. 2013. Identification of promising patents for technology transfers using triz evolution trends. *Expert systems with applications*, 40(2):736–743.

Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S Weld. 2024. Scideator: Human-llm scientific idea generation grounded in research-paper facet recombination. *arXiv preprint arXiv:2409.14634*.

Runtao Ren, Jian Ma, and Jianxi Luo. 2025. Large language model for patent concept generation. *Advanced Engineering Informatics*, 65:103301.

Harsimrat Singh Sandhawalia, Jose Antonio RODRIGUEZ SERRANO, Herve Poirier, and Gabriela Csurka. 2017. Vehicle classification from laser scanners using fisher and profile signatures. US Patent 9,683,836.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

L Siddharth and Jianxi Luo. 2024. Retrieval augmented generation using engineering design knowledge. *Knowledge-Based Systems*, 303:112410.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Achille Souili and Denis Cavallucci. 2012. Toward an automatic extraction of idm concepts from patents. In *CIRP Design 2012: Sustainable Product Development*, pages 115–124. Springer.

Achille Souili, Denis Cavallucci, and François Rousselot. 2015. A lexico-syntactic pattern matching method to extract idm-triz knowledge from on-line patent databases. *Procedia engineering*, 131:418–425.

Stefan Trapp and Joachim Warschat. 2024. Llm-based extraction of contradictions from patents. In *International TRIZ Future Conference*, pages 3–19. Springer.

Gangfeng Wang, Xitian Tian, Junhao Geng, Richard Evans, and Shengchuang Che. 2016. Extraction of principle knowledge from process patents for manufacturing process innovation. *Procedia Cirp*, 56:193–198.

Juanyan Wang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. 2024a. Patentformer: a novel method to automate the generation of patent applications. In *Proceedings of the 2024 conference on empirical methods in natural language processing: industry track*, pages 1361–1380.

Suyuan Wang, Xueqian Yin, Menghao Wang, Ruofeng Guo, and Kai Nan. 2024b. Evopat: A multi-llm-based patents summarization and analysis agent. *arXiv preprint arXiv:2412.18100*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Qizhi Xie and Qiang Liu. 2023. Application of triz innovation method to in-pipe robot design. *Machines*, 11(9):912.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Siyeong Yun, Woojin Cho, Chulhyun Kim, and Sungjoo Lee. 2022. Technological trend mining: identifying new technology opportunities using patent semantic analysis. *Information Processing & Management*, 59(4):102993.

Qihao Zhu and Jianxi Luo. 2022. Generative design ideation: a natural language generation approach. In *International Conference on-Design Computing and Cognition*, pages 39–50. Springer.

A Appendix

A.1 More Extraction Details

A.1.1 Schema

Each TRIZ case is stored as a JSON object with required fields for (i) system context, (ii) problem statement, (iii) contradiction structure, and (iv) resolution. Contradictions are normalized to an improve–worsen pair (`improving_text`, `worsening_text`) for both technical and physical contradictions. Each major field includes evidence spans that point back to the source segment text. We additionally support optional metadata fields, including `patent_ids` and TRIZ parameter/principle identifiers when explicitly present in the source.

A.1.2 Prompts

We summarize our prompts to extract cases for short as follows:

Segment detection prompt. Given a document segment, the detector predicts whether it contains at least one extractable TRIZ case (`case` vs. `no_case`). We provide the seed exemplars as in-context examples to calibrate the boundary.

Structured extraction prompt. For candidate segments, the extractor is instructed to output a JSON object (or list) that conforms to the schema, attach evidence spans for each contradiction and resolution, and abstain when key components (system, contradiction, resolution) are missing.

A.1.3 Validation and Deduplication

We apply two classes of automatic checks.

Schema validation. We verify JSON parseability; required keys and types; and controlled-vocabulary fields (e.g., contradiction type). Outputs that violate required fields are discarded.

Evidence grounding. We verify that evidence spans correspond to text in the processed segment (or a small local window, when used). Outputs missing evidence for contradiction or resolution are discarded.

Deduplication. We merge near-duplicate cases within the same document using similarity between normalized contradiction texts and resolution snippets, retaining the most complete object and merging complementary metadata when applicable.

A.1.4 Patent Corpus Details

Patent ID normalization We normalize patent identifiers into a consistent U.S. grant format (digits only) and deduplicate repeated records.

PatentsView retrieval For each normalized patent ID, we query PatentsView for a standardized set of fields (e.g., title, abstract, background/-summary when available, claims, CPC codes, grant date). Missing fields are recorded explicitly. We restrict the dataset to U.S. patents supported by PatentsView and discard non-U.S. references.

A.1.5 Human evaluation guidelines

We provide a annotation tool to the human experts and a detailed list on what to check in the cases as shown in Figure 4.

A.2 Additional dataset statistics

We release TRIZBENCH as a JSONL file (`triz_cases.jsonl`) containing one TRIZ case per line. Each case is represented by a fixed schema (Table 4) with four evidence-grounded components: *system context*, *problem setting*, *contradiction structure*, and *resolution*. Contradictions are stored as a list of conflict objects (Table 5) that include free-text fields describing the improving and worsening requirements (`improving_aspect_text`, `worsening_aspect_text`), an optional contradiction type label (`contradiction_type`), and optional mappings to TRIZ engineering parameters when available (e.g., `triz_improving_parameter_id/name`, `triz_worsening_parameter_id/name`). Solutions are recorded as free text (`solution_text`) and, when explicitly stated in the source, as a list of TRIZ inventive principles (`triz_principles`. To support grounded evaluation, each contradiction and principle entry includes an `evidence_text` field containing the supporting snippet from the source; when offset tracking is available, we additionally provide optional global `evidence_spans` that link extracted fields to approximate document locations. We use controlled vocabularies for fields such as `contradiction_type` to ensure consistent labeling across the corpus, while keeping the core descriptions in free text to preserve the original technical framing.

[3] In our current pipeline, evidence is primarily stored at the contradiction/principle level via `evidence_text`. Global `evidence_spans` are optionally populated when offset tracking is available.

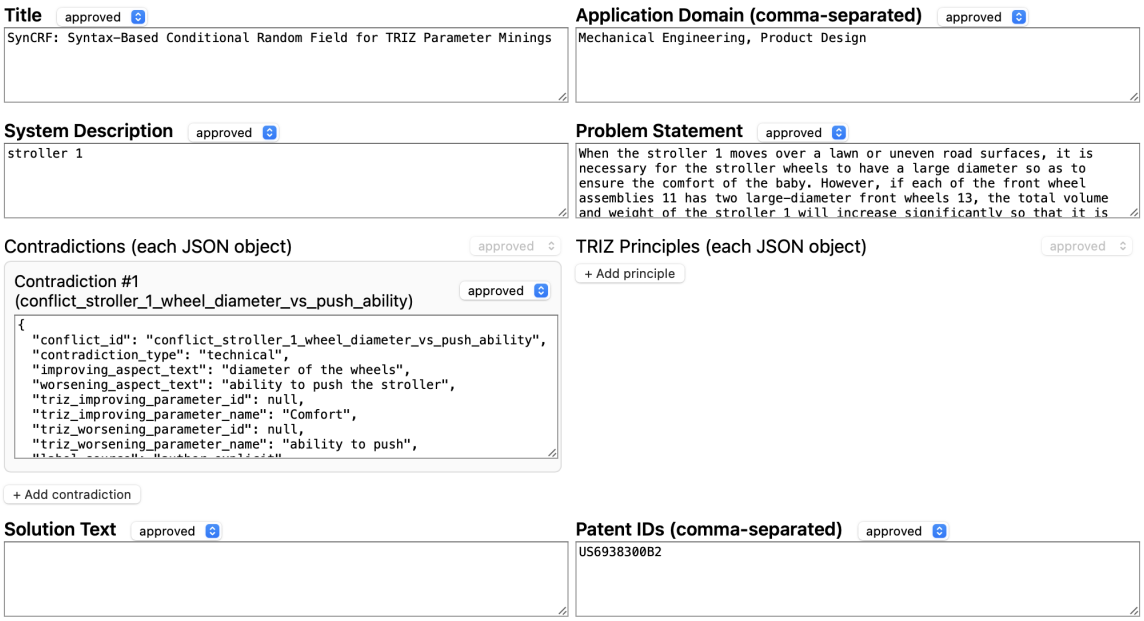


Figure 4: The annotation tool for cases evaluation.

Field	Type	Req.	Description / Example
`case_id`	string	Y	Unique case identifier.
`source_doc_id`	string	Y	Source document identifier.
`source_type`	string	Y	Source category (e.g., `journal_article`, `triz_journal_html`).
`title`	string	Y	Document title.
`year`	int	Y	Publication year.
`language`	string	Y	Language code (e.g., en).
`application_domain`	list[string]	N	Domain tags (e.g., Mechanical Engineering; Pharmacy Automation).
`system_description`	string	Y	System context and background (free text).
`problem_statement`	string	Y	Problem narrative motivating the contradiction(s).
`contradictions`	list[dict]	Y	List of extracted contradiction objects (Table 5).
`solution_text`	string	N	Solution mechanism described in the source (free text).
`triz_principles`	list[dict]	N	List of TRIZ inventive principles linked to the solution.
`patent_ids`	list[string]	N	Referenced patent identifiers if available.
`evidence_spans`	list[dict]	N	evidence spans (segment/page offsets when available).
`has_problem`	bool	Y	Presence indicator for problem statement.
`has_contradictions`	bool	Y	Presence indicator for contradiction list.
`has_solution`	bool	Y	Presence indicator for solution text.
`has_principles`	bool	Y	Presence indicator for TRIZ principles list.

Table 4: Top-level JSON schema for a TRIZ case in **TRIZBENCH**. "Req." denotes required fields.

Field	Type	Req.	Notes
`conflict_id`	string	Y	Unique ID within a case.
`contradiction_type`	string	Y	{`technical`, `physical`}.
`improving_text`	string	Y	Text span describing the desired improvement.
`worsening_text`	string	Y	Text span describing the trade-off/worsening.
`triz_improving_parameter_id`	int	N	Optional TRIZ engineering parameter ID.
`triz_improving_parameter_name`	string	N	Optional parameter name.
`triz_worsening_parameter_id`	int	N	Optional TRIZ engineering parameter ID.
`triz_worsening_parameter_name`	string	N	Optional parameter name.
`evidence_text`	string	Y	Evidence snippet supporting this contradiction.
`confidence`	float	N	Model confidence when available.

Table 5: Schema for contradiction objects in `contradictions`.

Major domain	# cases
TRIZ/Innovation/Management	218
Mechanical/Manufacturing	189
Computer/Communication	176

Table 6: Top 3 Major domain case counts.

CPC 3-digit class	# patents
A61	173
G06	85
Y10	83
H01	72
G01	66

Table 7: Top CPC 3-digit classes in the patent corpus.

A.2.1 Major domain mapping and coverage

The raw `application_domain` field contains fine-grained tags originating from heterogeneous sources. For presentation and analysis, we map these tags into **10** major domains using a deterministic keyword-based mapping: *Mechanical/Manufacturing*, *Electrical/Electronics*, *Computer/Communication*, *Transportation/Aerospace*, *Chemical/Materials*, *Biomedical/Pharma*, *Energy/Environment*, *Civil/Construction*, *Product/Consumer*, and *TRIZ/Innovation/Management*. A case may map to multiple macro domains when tags indicate cross-domain systems. Table 6 reports top 3 major domain frequencies.

A.2.2 Patent corpus and CPC coverage

The patent corpus consists of (i) **223** unique patents referenced by **148** cases in the extracted corpus, and (ii) an auxiliary set of **234** author-curated patents with TRIZ parameter/principle labels collected from TRIZ-focused resources, for a total of **429** unique patents. Across the union set, **257** patents (**56.2%**) include TRIZ parameter labels and **304** patents (**66.5%**) include at least one TRIZ principle label. We retrieve patent metadata and CPC assignments via PatentsView and observe coverage of **83** distinct 3-digit CPC classes (e.g., `G06`, `H04`, `A61`). Table 7 lists the most frequent CPC classes.

Patent text lengths. To contextualize input length for patent-centric benchmarks, the mean patent abstract length is **187** tokens and the mean first independent claim length is **216** tokens under our retrieval pipeline (token counts estimated from character length for model-agnostic reporting).

A.3 More Experiments Details

We include the prompts we used in contradiction prediction and principle prediction, and evidence constriction in grounded TRIZ reasoning. In addition, we also show more detailed results of contradiction prediction in Table 8. Note that, all experiments are reported as average results of three runs.

A.3.1 Contradiction prediction

In LoRA settings, we run 2 epochs with learning rate of $2e-4$, batch size as 1 and max sequence length as 4096 for all models.

For all experiments with prompts, we use the following prompt:

Prompt (Contradiction prediction)

```
SYSTEM:
You are an expert TRIZ analyst.
Extract contradictions and
output ONLY a valid JSON object.

USER:
Task: Identify and structure TRIZ contradictions
from the text.

Return ONLY a JSON object matching this schema
(keys must match; use null if unknown):


Rules:
1) Output MUST be valid JSON. No markdown.
No extra text.
2) If no contradiction is explicitly supported
by the text, set has_contradictions=false
and contradictions=[].
3) Output AT MOST 2 contradictions.
If multiple exist, choose the 2 most central and
explicitly stated.
4) improving_text and worsening_text MUST
be short phrases.
5) evidence_text MUST be a short quote from the
INPUT TEXT that supports the contradiction
   (not an explanation).
6) Parameter IDs are optional;
set to null unless the text explicitly
supports the mapping.

INPUT TEXT:
{input_text}
```

A.3.2 More results

Table 8 reports more results on different semantic matching thresholds on case predictions and also expanded patent-level ranking metrics. As expected, PairF1 decreases as the semantic threshold increases from τ=0.7 to τ=0.75 across all models, which means that exact contradiction phrasing is a major source of error even when predictions are semantically close. On patents, Hit@1 remains

low overall, but Hit@5 increases substantially, suggesting that many correct pairs appear in the candidate set but are poorly ranked. Notably, retrieval baselines are competitive at Hit@5 (PatentSBERTa 0.38), and LoRA achieves the best ranking performance (up to 0.48), reinforcing the value of semantic matching plus learned ranking under domain shift.

A.3.3 Principle Prediction

For all experiments with prompts, we use the following prompt:

Prompt (Principle Prediction)

```
SYSTEM:
You are an expert TRIZ analyst.
Given a problem and contradiction,
predict the most relevant TRIZ
inventive principles.
Return ONLY valid JSON.

USER:
Task: Predict the most relevant TRIZ inventive
principles (IDs 1..40) to resolve the
contradiction.
Return ONLY a JSON object with this schema:
{
  "principle_ids": [int, ...]
}

Rules:
1) Output must be valid JSON only.
2) principle_ids must be integers
in [1..40], unique, ranked best-first.
3) Output EXACTLY K IDs (no fewer).

TRIZ PRINCIPLES (1..40):
1. ...
2. ...
...
40. ...

INPUT:
{input_text}
```

A.3.4 GTR evidence construction

We construct evidence sets via retrieval over patent sentences. For each component $c \in \{\text{imp}, \text{wor}, \text{sol}\}$, we define anchors A^c from case fields, including a^{imp} for improving, a^{wor} for worsening, and case solution/principle evidence text for solution. We score each sentence by its maximum similarity to the anchors and take top-K: $\hat{E}^c = \text{TopK}\Big(\max_{\alpha \in A^c} \text{sim}(s_i, \alpha) \Big)$, where sim is PatentSBERTa embedding cosine similarity.

Model	Method	Case		Patent	
		PairF1 @ 0.7	**PairF1 @ 0.75**	**Hit@1**	**Hit@5**
Qwen3-7B	ZS	0.28	0.24	0.08	0.19
Qwen3-7B	FS	0.30	0.27	0.08	0.24
Qwen3-32B	ZS	0.27	0.26	0.10	0.28
Qwen3-32B	FS	0.32	0.27	0.11	0.32
LLaMA3.1-8B	ZS	0.25	0.22	0.04	0.21
LLaMA3.1-8B	FS	0.28	0.24	0.10	0.22
LLaMA3.1-70B	ZS	0.29	0.27	0.12	0.26
LLaMA3.1-70B	FS	0.31	0.26	0.14	0.31
Gemini-2.5-Pro	ZS	0.30	0.28	0.11	0.30
Gemini-2.5-Pro	FS	0.35	0.30	0.13	0.33
PatentSBERTa	Retrieval	–	–	0.19	0.38
BGE	Retrieval	–	–	0.15	0.35
Qwen3-7B	LoRA	0.42	0.38	0.22	0.48
LLaMA3.1-8B	LoRA	0.38	0.36	0.20	0.42

Table 8: Additional results on contradiction prediction. **PairF1** is semantic matching F1 for improve–worsen contradiction pairs at threshold $\tau = 0.7, 0.75$ (case-level). **Hit@k** reports whether a gold pair appears among the model's top-1,5 ranked predicted pairs (patent-level).