How can a transformer-based mediation layer incorporate formal verification methods to guarantee that natural language instructions do not violate the robot's physical safety boundaries?

Transformer-based mediation layers can incorporate formal verification methods by translating natural language instructions into temporal logic specifications (such as Linear Temporal Logic or Signal Temporal Logic) through chain-of-thought reasoning and iterative refinement, then verifying these specifications against safety constraints using model checking, Control Barrier Functions, or reachability analysis before execution—an approach that reduces unsafe plan execution by up to 90% in controlled settings, though complete formal guarantees for arbitrary instructions in open-ended real-world environments remain beyond current capabilities due to computational scalability limitations and the simulation-to-reality gap.

Abstract

This systematic review of 80 sources reveals that transformer-based mediation layers can incorporate formal verification methods through three primary architectural patterns: pipeline architectures where transformer outputs undergo separate verification stages , integrated systems embedding verification within transformer inference , and mediation layer approaches generating intermediate formal specifications for downstream planners . Linear Temporal Logic (LTL) represents the most widely adopted verification method , complemented by Signal Temporal Logic for continuous dynamics , Control Barrier Functions for real-time guarantees , and conformal prediction for uncertainty quantification . Translation from natural language to formal specifications employs chain-of-thought reasoning , equivalence voting achieving up to 98% accuracy , and iterative refinement with formal feedback improving specification compliance from 60% to over 90% . These approaches yield substantial safety improvements: RoboGuard reduces unsafe plan execution from 92% to below 2.5% , SAFER achieves 77.5% reduction in safety violations , and SafePlan demonstrates 90.5% reduction in harmful task prompt acceptance .

However, the evidence reveals fundamental trade-offs limiting current guarantees. Formal verification tools scale only to networks with hundreds of neurons while modern transformers contain millions of parameters , necessitating approximation strategies such as relaxation-based verification or probabilistic bounds . Real-world deployment consistently shows degraded performance compared to simulation due to imperfect perception , and current approaches cannot adequately capture time-bounded behaviors . The synthesis indicates that transformer-based mediation layers can effectively guarantee physical safety when safety requirements are expressible in tractable temporal logics, computational resources permit real-time verification, environmental uncertainty is bounded, and human oversight remains available for edge cases . Complete formal guarantees for arbitrary natural language instructions in open-ended environments remain beyond current capabilities.

Paper search

We performed a semantic search using the query ”How can a transformer-based mediation layer incorporate formal verification methods to guarantee that natural language instructions do not violate the robot's physical safety boundaries?” across over 138 million academic papers from the Elicit search engine, which includes all of Semantic Scholar and OpenAlex.

We retrieved the 489 papers most relevant to the query.

Screening

We screened in sources based on their abstracts that met these criteria:

- **Robotic NLP Systems**: Does this study involve robotic systems that process natural language commands or instructions AND incorporate transformer-based architectures, neural language models, or similar deep learning approaches for natural language understanding in robotics?
- **Formal Verification Methods**: Does this study implement, propose, or evaluate formal verification methods, safety verification techniques, or mathematical proof systems in robotic contexts?
- **Physical Safety Focus**: Does this research address physical safety constraints, safety boundaries, collision avoidance, or harm prevention in robotic systems?
- **ML-Formal Methods Integration**: Does this study examine the integration or combination of machine learning approaches with formal methods or safety verification?
- **Safety-Critical Applications**: Does this research focus on safety-critical robotics applications such as industrial robots, autonomous vehicles, service robots, or human-robot interaction scenarios?
- **Study Type and Quality**: Is this study an experimental study, theoretical paper, case study, systematic review, or meta-analysis that contains technical content or empirical evaluation (not merely opinion pieces, editorials, or purely speculative articles)?
- **Physical Safety Scope**: Does this study address physical safety considerations (rather than focusing solely on software safety or cybersecurity without physical safety aspects)?
- **Robotics Relevance**: Does this research maintain relevance to robotics applications (rather than being limited to general natural language processing without robotics applications OR formal verification methods unrelated to robotics, safety systems, or real-time constraints)?

We considered all screening questions together and made a holistic judgement about whether to screen in each paper.

Data extraction

We asked a large language model to extract each data column below from each paper. We gave the model the extraction instructions shown below for each column.

- **Architecture Design**:

  Extract the specific technical architecture for integrating transformers with formal verification including:

- How the transformer and verification components are connected
- Whether it's a mediation layer, pipeline, or integrated system
- Multi-level verification approaches (semantic, plan, trajectory, etc.)
- Data flow between components
- Real-time vs offline verification

- **Formal Verification Methods**:

  Document all formal verification techniques used including:

- Specific formal methods (Linear Temporal Logic, model checking, etc.)
- Mathematical frameworks or logics employed
- Verification algorithms or tools used
- Completeness and soundness guarantees
- Computational complexity considerations

- **Safety Constraint Definition**:

  Extract how physical safety boundaries are specified and represented including:

- Method for defining safety constraints (temporal logic formulas, invariants, etc.)
- Types of physical limitations covered (spatial boundaries, force limits, collision avoidance, etc.)
- How constraints are encoded from domain knowledge
- Granularity and specificity of safety specifications
- Adaptability to different robot platforms

- **NL-to-Formal Translation**:

  Document the process of translating natural language instructions to formal specifications including:

- Parsing and semantic analysis methods
- How ambiguity and uncertainty are handled
- Intermediate representations used
- Validation of translation accuracy
- Handling of complex or conditional instructions
- Error detection in translation process

- **Transformer Component**:

  Extract details about the transformer architecture and its role including:

- Specific transformer model and modifications
- Input types processed (text, visual, proprioceptive, etc.)
- Output format and structure
- Integration with robot control systems
- History/memory mechanisms
- Training methodology and datasets used

- **Safety Guarantees Achieved**:

  Document the safety guarantees and verification results including:

- Types of safety violations prevented
- Completeness of safety coverage
- False positive/negative rates
- Quantitative safety metrics achieved
- Comparison with baseline or previous methods
- Real-world vs simulation performance
- Any safety failures or edge cases identified

- **Evaluation Methods**:

  Extract how the system was evaluated including:

- Experimental environments (simulation platforms, real robots)
- Test scenarios and task complexity
- Safety requirement diversity
- Performance metrics measured
- Baseline comparisons

- Statistical analysis methods
- Duration and scale of testing

- **Limitations and Challenges**:

  Document identified limitations and technical challenges including:

- Computational overhead and scalability issues
- Types of instructions or scenarios not handled
- Verification incompleteness or approximations
- Integration difficulties
- Deployment constraints
- Future work needed
- Trade-offs between safety and performance

Report

Due to the limitations of the AI model, we are only able to process 80 sources while writing a report. This report was written using the 80 sources that had the highest screening scores out of the 469 sources that we screened in and extracted data from.

Characteristics of Included Studies

The reviewed literature encompasses 80 sources investigating the integration of transformer-based systems with formal verification methods for robotic safety. The studies span multiple domains including autonomous navigation, robotic manipulation, multi-robot coordination, and autonomous driving.

Study	Full text retrieved?	Primary Focus	Architecture Type
Ahmad Hafez et al., 2025	Yes	LLM-controlled robot safety via reachability analysis	Pipeline with safety layer
Zachary Ravichandran et al., 2025	Yes	Safety guardrails for LLM-enabled robots	Two-stage guardrail architecture
Yunhao Yang et al., 2024	Yes	Joint verification and refinement of LMs	Automaton-based verification pipeline
S. Zhan et al., 2025	Yes (abstract only)	Multi-level formal safety evaluation	Multi-level verification pipeline
Ziyi Yang et al., 2023	Yes	Safety constraint enforcement via LTL	Integrated system with safety chip
Abdulrahman Althobaiti et al., 2024	Yes	LLM and knowledge graph safety layer	Pipeline with mediation layer
Leonardo Santos et al., 2024	Yes	Online safety representation updates	Integrated VLM-based system
Ike Obi et al., 2025	Yes	Multi-component safety framework	Pipeline with COT reasoners
Ziming Wang et al., 2024	Yes (abstract only)	Cross-layer sequence supervision	Cross-layer safety supervisor

Study	Full text retrieved?	Primary Focus	Architecture Type
Kumar Manas et al., 2023	Yes (abstract only)	NL rule formalization for intelligent vehicles	Not mentioned
Lukas Brunke et al., 2024	Yes (abstract only)	Semantic safety filtering	Control barrier certification
A. Khan et al., 2025	Yes	Safety-aware task planning framework	Multi-LLM framework with Safety Agent
Qian Meng et al., 2025	Yes (abstract only)	LLM-based controller repair	Pipeline architecture
Yunhao Yang et al., 2025	Yes (abstract only)	Fine-tuning-free planning via formal feedback	Not mentioned
Yi Wu et al., 2024	Yes	Safe and efficient task planning	Integrated system with constrained decoding
Wanjing Huang et al., 2025	Yes (abstract only)	Graphormer-enhanced risk-aware planning	Graphormer with LLM
P. Sermanet et al., 2025	Yes	Robot constitution generation	Constitutional AI approach
Jiabao Ji et al., 2025	Yes (abstract only)	Collision-aware multi-robot control	RLVR integration
Haoyu Wang et al., 2025	Yes	Proactive runtime enforcement	Four-stage pipeline with DTMC
Teun van de Laar et al., 2024	Yes	NL-driven robot via formal specifications	Transformer with STL verification
E. Kaigom et al., 2023	Yes (abstract only)	Natural robot guidance using transformers	Not mentioned
J. Rosser et al., 2023	Yes (abstract only)	Dialogue-based ambiguity resolution	Not mentioned
Benedict Quartey et al., 2024	Yes (abstract only)	Verifiable robot instruction following	Foundation models with temporal logic
Seif Ismail et al., 2024	Yes	NL architecture for optimal control	LLM integrated with MPC
Maximilian Tolle et al., 2025	Yes (abstract only)	Safe robot foundation models	Safety layer with ATACOM
Jun Wang et al., 2024	Yes (abstract only)	Safe multi-robot planning with conformal prediction	Decentralized LLM planner
Jeremy Siburian et al., 2025	Yes (abstract only)	Grounded VLM interpreter for TAMP	Hybrid planning framework
Amir Bayat et al., 2025	Yes	LLM-enhanced symbolic control	Code Agent with Checker Agent
Jiayi Pan et al., 2023	Yes (abstract only)	Data-efficient NL to LTL translation	LLM with constrained decoding
Sara Mohammadinejad et al., 2022	Yes (abstract only)	Interactive learning using STL	Semantic parsing with transformers
Sathwik Karnik et al., 2024	Yes	Embodied red teaming	VLM-based evaluation
Parv Kapoor et al., 2025	Yes (abstract only)	Constrained decoding for robotics	STL-based framework

Study	Full text retrieved?	Primary Focus	Architecture Type
Zeyu Feng et al., 2024	Yes (abstract only)	Temporally-extended constraint satisfaction	Diffusion with LTL guidance
A. Benton et al., 2023	Yes (abstract only)	Verifiable learned behaviors	Motion primitive composition
J. Liu et al., 2023	Yes (abstract only)	NL to temporal robot specification	Modular LLM system
J. S. Park et al., 2017	Yes (abstract only)	Realtime motion plans from NL	Dynamic Grounding Graph
Zitong Bo et al., 2025	Yes (abstract only)	Reinforced embodied planning	VLM with verifiable reward
Yong Qi et al., 2024	Yes (abstract only)	Safety control with knowledge graphs	LLM with EKGs
Junhui Huang et al., 2025	Yes (abstract only)	Geometry-aware trajectory reshaping	VLM with LLM constraints
Shaojun Xu et al., 2024	Yes (abstract only)	Multi-robot hierarchical temporal logic	Two-step LLM process
Behrad Rabiei et al., 2025	Yes	LTL code generation for task planning	Modular LTL translation
Devesh Nath et al., 2025	Yes (abstract only)	Formal safety verification of GMPs	NNV-based verification
Aishan Liu et al., 2025	Yes	Safety benchmark for embodied agents	Two-component semantic adapter
Yunhao Yang et al., 2023	Yes	Fine-tuning LMs using formal feedback	Automaton-based pipeline
Daniel Ekpo et al., 2024	Yes	Scene graphs for verifiable planning	Iterative planning pipeline
Hangtao Zhang et al., 2024	Yes (abstract only)	Jailbreaking embodied LLMs	Not mentioned
Borong Zhang et al., 2025	Yes (abstract only)	Safety alignment of VLA models	Constrained learning framework
Dan BW Choe et al., 2025	Yes	LLM-to-TL framework for cooperation	BNF-constrained LLM with MILP
William English et al., 2024	Yes	Neuro-symbolic navigational planner	Integrated feedback loop
Ana Davila et al., 2025	Yes	LLM ambiguity detection in surgery	Ensemble LLM evaluators
Sabit Hassan et al., 2024	Yes (abstract only)	Multimodal safety dialogue	Coherence-driven dialogue system
Ana Davila et al., 2025a	Yes (abstract only)	Affordance-based disambiguation	Dual-set conformal prediction
J. S. Park et al., 2017a	Yes (abstract only)	Motion plans from attribute-based NL	Dynamic Constraint Mapping
William Xie et al., 2025	Yes	Dual-use dilemma in physical reasoning	VLM safeguarding evaluation

Study	Full text retrieved?	Primary Focus	Architecture Type
Milan Ganai et al., 2025	Yes (abstract only)	Real-time OOD failure prevention	Multi-modal reasoning framework
Jun Wang et al., 2025	Yes (abstract only)	Conformal NL to LTL translation	Iterative QA with CP
Parv Kapoor et al., 2024	Yes	Logically constrained transformers	STL-integrated transformer
Yue Meng et al., 2023	Yes (abstract only)	Control Barrier Transformer	Causal transformer with CBF
Marta Skreta et al., 2023	Yes	Instruction guided task programming	Generator-verifier iterative loop
Tsung-Yen Yang et al., 2020	Yes (abstract only)	Safe RL with NL constraints	Constraint interpreter network
Yilin Wu et al., 2025	Yes	VLM-in-the-loop policy steering	Decoupled prediction-evaluation
Junle Li et al., 2025	Yes	Automatic safety-compliant LTL generation	Self-supervised verification
Vanya Cohen et al., 2024	Yes (abstract only)	Survey of robotic language grounding	Not applicable (survey)
Zhendong Chen et al., 2025	Yes	NL-to-robotic language translation	RSL compiler verification
Kaiqu Liang et al., 2024	Yes (abstract only)	Introspective planning	Uncertainty-aware planning
Jason Liu et al., 2022	Yes (abstract only)	NL to LTL translation	Neural machine translation
L. Guan et al., 2024	Yes (abstract only)	VLM as behavior critics	VLM verification framework
Nishanth Kumar et al., 2024	Yes	Open-world TAMP via VLM constraints	VLM-TAMP mediation
Jun Wang et al., 2024a	Yes	Probabilistically correct multi-robot planning	Decentralized CP-based planner
Kumar Manas et al., 2024	Yes (abstract only)	Low-resource temporal knowledge	CoT-based LTL generation
Kaiqu Liang et al., 2024a	Yes (abstract only)	Introspective planning refinement	Uncertainty-aware LLM
Yunhao Yang et al., 2023a	Yes (abstract only)	Multimodal pretrained models	Automaton-based controller
Mani Amani et al., 2025	Yes (abstract only)	Digital twin-guided path planning	Beta-Bernoulli fusion
Kumar Manas et al., 2024a	Yes	Traffic rules to MTL formalization	CoT in-context learning
Aladin Djuhera et al., 2025	Yes (abstract only)	LLM-based constraint generation	Executable Python functions

Study	Full text retrieved?	Primary Focus	Architecture Type
Vasileios Manginas et al., 2025	Yes	Probabilistic neuro-symbolic verification	Relaxation-based verification
Parv Kapoor et al., 2025a	Yes	Embedding Temporal Logic	ETL specification framework
Zirui Song et al., 2025	Yes	Reinforcement learning for manipulation	RLVR framework
Chen Ding et al., 2025	Yes (abstract only)	Robotic instruction optimization	SC-RAG-CoT framework
Akkamahadevi Hanni et al., 2023	Yes (abstract only)	Safe explicable robot planning	Not mentioned

The studies demonstrate considerable diversity in architectural approaches, with pipeline architectures being most common (35 studies), followed by integrated systems (22 studies), and hybrid approaches combining multiple paradigms (15 studies). Eight studies did not provide sufficient architectural detail.

Architecture Designs for Integrating Transformers with Formal Verification

Pipeline Architectures

The predominant architectural pattern involves sequential processing where transformer outputs are verified before execution. RoboGuard employs a two-stage guardrail architecture where a root-of-trust LLM generates safety specifications using chain-of-thought reasoning, which are then verified through temporal logic control synthesis . Similarly, the framework by Ahmad Hafez et al. connects an LLM generating plans to a safety layer that verifies and adjusts those plans using data-driven reachability analysis .

Several studies implement multi-level verification pipelines. Sentinel uses a verification pipeline operating at semantic, plan, and trajectory levels, where natural language safety requirements are first formalized into temporal logic formulas, then action plans are verified against these formulas, and finally execution trajectories undergo detailed specification checking . The cross-layer sequence supervision mechanism proposed by Ziming Wang et al. employs a safety supervisor that provides closed-loop correction at the task planning layer while introducing virtual obstacles for motion planning .

Integrated Systems

Several approaches tightly couple transformer components with verification mechanisms. The "Safety Chip" architecture implements an integrated system where a queryable safety constraint module based on Linear Temporal Logic connects a language understanding system with formal verification methods, enabling NL to LTL translation, safety violation reasoning, and action pruning within a unified framework . Pro2Guard uses a four-stage pipeline involving trace collection, abstraction, Discrete-Time Markov Chain learning, and runtime verification, with probabilistic model checking for real-time safety enforcement .

The SELP framework demonstrates an integrated approach combining equivalence voting, constrained decoding, and domain-specific fine-tuning, where LTL specifications serve as intermediate representations that enable constrained decoding to generate safe plans . PASTEL integrates Signal Temporal Logic specifications directly with autoregressive transformer models using cross attention mechanisms to ensure predictions adhere to formal specifications .

Mediation Layer Approaches

A distinct category of architectures employs explicit mediation layers between language models and robot control systems. VernaCopter uses a planning assistant that translates natural language commands into STL specifications, with syntax checking (SynCheQ) and semantic alignment checking (SemCheQ) components providing verification . The SAFER framework employs a Safety Agent operating alongside the primary task planner, providing safety feedback while Control Barrier Functions ensure safety guarantees .

OWL-TAMP deploys VLMs within TAMP systems by having them generate discrete and continuous language-parameterized constraints that augment traditional manipulation constraints . The Code Agent and Checker Agent architecture uses a feedback loop where the Code Agent interprets natural language descriptions while the Checker Agent verifies generated code against original specifications .

Formal Verification Methods Employed

Temporal Logic Specifications

Linear Temporal Logic (LTL) represents the most widely adopted formal verification approach, employed in 28 studies. The method enables precise specification of temporal properties including sequencing, liveness, and safety invariants . LTL formulas are typically converted to Büchi automata for verification, enabling formal model checking against safety specifications .

Signal Temporal Logic (STL) provides additional capabilities for specifying temporal constraints with real-valued time bounds, making it suitable for continuous robot dynamics . The VernaCopter system demonstrates STL's utility in providing rigorous task descriptions while maintaining tractability for motion planning optimization .

Metric Temporal Logic (MTL) extends temporal specifications with quantitative timing constraints, particularly useful for traffic rule formalization and autonomous driving applications . The TR2MTL framework achieves domain-agnostic translation from natural language traffic rules to MTL specifications using chain-of-thought prompting .

Reachability Analysis and Control Theory

Hamilton-Jacobi reachability analysis provides mathematically rigorous safety guarantees by computing backward reachable sets representing states from which unsafe conditions are unavoidable . This approach enables policy-agnostic safety controllers that can be updated online as new constraints are introduced through language feedback .

Control Barrier Functions (CBFs) offer another principled approach to safety verification, ensuring forward invariance of safe sets through constrained optimization . The SAFER framework integrates CBFs with LLM-based planning to provide theoretical safety guarantees while modifying nominal controllers in real-time .

Data-driven reachability analysis extends traditional approaches by using historical data to construct reachable sets for robot-LLM systems without requiring explicit analytical models . This method provides rigorous safety guarantees while accommodating the inherent uncertainty of language model outputs.

Probabilistic and Statistical Verification

Conformal prediction offers distribution-free uncertainty quantification for black-box language models, enabling probabilistic safety guarantees . This approach allows systems to reason about inherent uncertainty in LLM-generated outputs and proceed with translation only when sufficiently confident .

Probabilistic model checking using PRISM verifies stochastic system behavior against Probabilistic Computation Tree Logic (PCTL) specifications . Pro2Guard uses Probably Approximately Correct (PAC) bounds to ensure statistical reliability of learned Discrete-Time Markov Chains modeling agent behavior .

Neural Network Verification

Several approaches leverage neural network verification (NNV) tools to certify closed-loop safety of learned policies . However, NNV tools currently scale only to networks with a few hundred neurons, presenting significant challenges for modern generative motion planners containing millions of parameters .

Verification Method	Studies Using	Key Advantages	Primary Limitations
Linear Temporal Logic	28	Precise temporal specifications, well-established theory	May not suit continuous dynamics
Signal Temporal Logic	7	Real-valued timing, robustness semantics	Computational complexity for optimization
Control Barrier Functions	4	Real-time guarantees, continuous systems	Requires explicit safe set definition
Reachability Analysis	3	Formal guarantees, handles uncertainty	Computational overhead for complex systems
Conformal Prediction	5	Distribution-free, black-box compatible	Requires calibration data
Probabilistic Model Checking	2	Handles stochasticity	Cannot capture time-bounded behaviors

Safety Constraint Definition and Representation

Temporal Logic-Based Constraints

The predominant method for defining safety constraints employs temporal logic formulas that specify invariants, temporal dependencies, and timing constraints. Studies using LTL encode spatial boundaries, collision avoidance requirements, and task sequencing constraints as formal specifications . The Sentinel framework demonstrates how natural language safety requirements can be formalized into temporal logic formulas that precisely capture state invariants and temporal dependencies .

SafePlan uses atomic predicates to represent world states, enabling structured specification of invariants, preconditions, and postconditions that must hold during task execution . The framework includes synonym handling mechanisms to normalize object names, addressing semantic ambiguity in natural language constraint specifications .

Physical Safety Boundaries

Multiple studies address specific physical limitations including spatial boundaries, force limits, and collision avoidance . The SAFER framework encodes comprehensive physical constraints including joint position limits, joint velocity limits, torque limits, obstacle avoidance, operational space limitations, singularity avoidance, and collision avoidance .

Control Barrier Functions provide a mathematically principled approach to defining safe sets, with constraints encoded as barrier functions that ensure the system remains within designated safety boundaries . The approach allows both global constraints applicable to all operations and step-specific constraints tailored to particular task phases.

Semantic and Contextual Safety

Beyond physical constraints, several studies address semantic safety including unsafe spatial relationships, behaviors, and poses . The semantic safety filter framework by Brunke et al. combines semantically unsafe conditions inferred by large language models with geometrically defined constraints for environment-collision and self-collision avoidance .

Vision-language models enable dynamic safety constraint updating based on natural language feedback and visual observations . This approach handles inherently personal, context-dependent constraints that can only be identified at deployment time, such as fragile objects or expensive surfaces .

Constraint Type	Representation Method	Example Studies	Adaptability
Spatial boundaries	LTL/STL predicates		Environment-specific
Collision avoidance	CBF, reachability sets		Dynamic updating
Force limits	CBF constraints		Robot-specific
Semantic constraints	VLM inference		Context-dependent
Temporal dependencies	LTL formulas		Task-specific

Natural Language to Formal Specification Translation

Parsing and Semantic Analysis

The translation from natural language to formal specifications employs diverse parsing strategies. Lang2LTL uses pretrained large language models to extract referring expressions from natural language commands, ground expressions to real-world landmarks, and translate commands into LTL task specifications . The modular approach achieves 88.4% accuracy in translating challenging LTL formulas across unseen environments .

Chain-of-thought reasoning has emerged as an effective technique for guiding step-by-step translation of complex natural language instructions . TR2MTL uses chain-of-thought in-context learning to decompose traffic rules into subtasks, enabling robust reasoning about conditional instructions .

Semantic role labeling combined with soft rule-based selection restrictions enables extraction of predicates, arguments, and temporal aspects from natural language rules . This approach provides implicit explanations of output by showing intermediate reasoning steps, enhancing interpretability of the translation process.

Intermediate Representations

Multiple studies employ structured intermediate representations to bridge the gap between natural language and formal specifications. Abstract Syntax Trees (ASTs) and Finite State Automata (FSAs) serve as intermediate representations enabling formal verification of translated specifications . The Exe2FSA algorithm converts executable plans generated by language models into automaton-based representations suitable for model checking .

Hierarchical Task Trees capture logical and temporal relations between sub-tasks, enabling translation of complex multi-step instructions into hierarchical LTL specifications . This representation simplifies planning while remaining straightforward to derive from human instructions.

Scene graphs provide an intermediate representation that captures object-level details as symbolic graphs, enabling constraint checking through graph operations . This approach allows quick validation of action feasibility while maintaining interpretability.

Handling Ambiguity and Uncertainty

Ambiguity in natural language poses significant challenges for formal translation. Equivalence voting addresses this by generating and sampling multiple LTL formulas from natural language commands, grouping equivalent formulas, and selecting the majority group as the final specification . This approach improves translation accuracy from 88.4% to 98.0% on benchmark datasets .

User-in-the-loop clarification provides an interactive approach to ambiguity resolution. DIALOGUESTL uses semantic parsing combined with user demonstrations to predict correct STL formulas from often ambiguous natural language descriptions . The approach is efficient, scalable, and robust with high accuracy using few demonstrations.

Conformal prediction enables uncertainty-aware translation by assessing confidence in LLM-generated answers . When uncertainty exceeds a threshold, the system can request clarification rather than proceeding with potentially incorrect translations.

Validation and Error Detection

Multiple validation mechanisms ensure translation accuracy. Syntax checking verifies that generated specifications conform to formal grammar requirements . The AutoSafeLTL framework implements a six-step generation strategy with syntactic and semantic checks to validate translation accuracy .

Semantic alignment checking ensures that translated specifications accurately reflect the original natural language intent . VernaCopter's SemCheQ component analyzes whether STL specifications align with task descriptions, detecting and correcting semantic errors.

Iterative refinement with formal feedback improves translation quality through multiple passes. CLAIRIFY uses verifier-assisted iterative prompting where syntax and constraint violations are fed back to the language model for correction . The process continues until a valid, executable plan is generated.

Transformer Components and Integration

Model Selection and Modifications

The reviewed studies predominantly employ large-scale pretrained language models, with GPT-4 family models being most common . Modifications typically involve fine-tuning for domain-specific tasks rather than architectural changes. SELP fine-tunes CodeLlama2-7b and Llama2-7b for LTL translation and planning using negative log-likelihood loss .

Smaller models demonstrate potential when properly constrained. Constraint-aware small LLMs (Qwen2.5-3B-Instruct, Qwen3-4B) outperform larger models without constraints when trained with reinforcement learning using verifiable rewards . This finding suggests that formal verification integration may reduce dependence on model scale.

Vision-language models enable processing of multimodal inputs for context-aware safety reasoning. OWLv2 VLM processes RGB-D images alongside natural language commands to update safety constraint representations . The FOREWARN framework adapts the Llama-3.2-11B-Vision-Instruct model by replacing observation tokenization with a world model's encoder to enable latent state reasoning .

Integration with Robot Control Systems

Integration approaches range from direct plan generation to constraint specification for downstream planners. NARRATE demonstrates layered integration where LLMs frame constraints and objective functions as mathematical expressions subsequently used in Model Predictive Control . This approach maintains interpretability while enabling flexible natural language control.

Several studies integrate transformers with motion planning through intermediate formal specifications. LTL formulas generated by LLMs are converted to Büchi automata and combined with semantic occupancy maps for motion planning . The resulting paths satisfy natural language instructions while avoiding collisions with mapped obstacles.

Real-time integration requires efficient inference and verification. PASTEL achieves online safety enforcement by using cross-attention mechanisms to ensure model predictions attend to specification tokens during autoregressive inference . This approach enables trajectory generation that satisfies STL specifications without requiring offline verification.

Safety Guarantees and Verification Results

Quantitative Safety Improvements

The reviewed studies report substantial improvements in safety metrics through formal verification integration. RoboGuard reduces execution of unsafe plans from 92% to below 2.5% without compromising performance on safe plans . In real-world experiments, the system prevents 100% of adversarial attacks while maintaining task completion capability .

SAFER achieves 77.5% reduction in safety violations with DeepSeek-r1 and 47% reduction with GPT-4o compared to unguarded baselines . SafePlan demonstrates 90.5% reduction in harmful task prompt acceptance while maintaining reasonable acceptance of safe tasks .

Fine-tuning with formal verification feedback improves specification compliance from 60% to over 90% . The joint verification and refinement approach achieves 30% improvement in probability of generating plans that meet task specifications .

Study	Baseline Safety	Verified Safety	Improvement	Evaluation Context
RoboGuard	8% (92% unsafe)	97.5%+	89.5%+	Adversarial attacks

Study	Baseline Safety	Verified Safety	Improvement	Evaluation Context
SAFER	Variable	77.5% reduction	Significant	Complex long-horizon tasks
SafePlan	Not specified	90.5% reduction	Significant	Harmful prompt filtering
SELP	Baseline planners	+10.8% safety rate	10.8%	Drone navigation
Fine-tuning with formal feedback	60%	90%	30%	Autonomous driving

Real-World Validation

Multiple studies validate safety guarantees on physical robot systems. Ahmad Hafez et al. demonstrate their safety assurance framework on a JetRacer in a Cyber-Physical Systems laboratory environment . The system ensures collision avoidance while navigating to specified goals under LLM control.

VernaCopter achieves 100% goal-reaching and collision-free rates in tested scenarios, significantly outperforming conventional NL-prompting-based planners . The formal verification approach eliminates unsafe plan execution while maintaining task completion capability.

NARRATE demonstrates successful real-world deployment on Franka Emika Panda and custom manipulator platforms, though collision rates increase in real-world settings due to imperfect perception . This highlights the importance of robust perception for maintaining verified safety properties during deployment.

Comparison with Baseline Approaches

Studies consistently demonstrate improvements over unverified baselines. SELP outperforms state-of-the-art planners by 10.8% in safety rate for drone navigation and 20.4% for robot manipulation tasks . NSP produces paths that are 19-77% shorter than state-of-the-art neural approaches while achieving 90.1% valid path generation .

Pro2Guard outperforms AgentSpec in runtime efficiency, probabilistic explainability, and engineering effort while achieving 100% prediction of traffic law violations and collisions in autonomous driving scenarios . The system can enforce safety on up to 93.6% of unsafe tasks in embodied agent domains.

The PASTEL approach achieves 74.3% higher specification satisfaction compared to baseline PACT models that lack formal constraint integration . This improvement demonstrates the value of incorporating temporal logic specifications directly into transformer architectures.

Synthesis: Reconciling Divergent Approaches

The literature reveals significant heterogeneity in how transformer-based systems incorporate formal verification for safety guarantees. This diversity reflects fundamental trade-offs between verification completeness, computational efficiency, and practical deployability.

Architectural Patterns and Their Implications

Pipeline architectures with separate verification stages predominate (approximately 44% of studies), offering modularity and clear separation of concerns . This approach enables use of established verification tools without modifying transformer architectures but may introduce latency and limit real-time adaptation. Studies reporting highest safety improvements (e.g., RoboGuard's 89.5%+ improvement) typically employ this pattern.

Integrated systems that embed verification within transformer inference (approximately 28% of studies) achieve tighter coupling but require specialized architectures . PASTEL's direct integration of STL specifications with transformer attention mechanisms exemplifies this approach, achieving 74.3% improvement in specification satisfaction . However, this approach limits compatibility with off-the-shelf models.

Mediation layer approaches that generate intermediate formal specifications provide flexibility while maintaining verifiability . These architectures enable use of existing motion planners and verification tools while leveraging transformer capabilities for natural language understanding.

Temporal Logic Selection and Application

The choice between LTL, STL, and MTL reflects different safety requirement characteristics. LTL suits discrete temporal specifications and enjoys mature verification tool support , but may not adequately capture continuous dynamics . STL provides quantitative semantics suitable for continuous systems but increases computational complexity. MTL offers timing constraint specification critical for real-time applications like autonomous driving .

Studies achieving highest translation accuracy employ multiple verification passes. Equivalence voting with chain-of-thought reasoning achieves 98.0% accuracy , while single-pass approaches typically achieve 75-90% . This suggests that verification confidence correlates with translation redundancy.

The Scalability-Completeness Trade-off

A fundamental tension exists between verification completeness and computational scalability. Formal methods based solely on model checking face computational explosion with large state spaces . Neural network verification tools scale only to hundreds of neurons while modern transformers contain millions of parameters .

Studies address this through approximation strategies. Relaxation-based verification scales exponentially better than solver-based solutions while maintaining soundness guarantees . Conformal prediction provides distribution-free uncertainty bounds without requiring complete state enumeration . Data-driven reachability analysis constructs safety guarantees from historical data rather than analytical models .

The practical implication is that current systems provide probabilistic rather than absolute safety guarantees in complex scenarios. Studies reporting highest safety rates typically evaluate in constrained environments with limited state spaces . Deployment in open-ended real-world settings remains challenging.

Real-World Deployment Considerations

Studies evaluating both simulation and real-world performance consistently report degradation in physical settings. NARRATE demonstrates increased collision rates in real-world deployment due to imperfect perception . This suggests that verified safety properties may not transfer completely across the simulation-to-reality gap.

Several architectural features enhance real-world robustness. Online constraint updating enables adaptation to deployment-time observations . Proactive risk prediction allows safety intervention before violations occur . Conformal prediction-based uncertainty quantification enables seeking clarification when confidence is insufficient .

However, computational overhead remains a barrier. High inference latency of large vision-language models necessitates manual intervention policies for time-critical scenarios . Systems achieving real-time performance typically employ smaller models or pre-computed verification results .

Emerging Consensus and Remaining Gaps

Despite methodological diversity, the literature converges on several principles. First, formal specification of safety requirements before execution outperforms post-hoc constraint checking . Second, multi-level verification (semantic, plan, trajectory) provides more robust guarantees than single-level approaches . Third, iterative refinement with formal feedback improves translation accuracy beyond single-pass methods .

Key gaps remain. Current approaches do not adequately handle time-bounded behaviors requiring real-time guarantees . Integration with vision-language-action models for end-to-end verification is nascent . Verification of multi-robot coordination under uncertainty lacks mature solutions . Security against adversarial manipulation of formal specifications themselves is underexplored .

The synthesis indicates that transformer-based mediation layers can effectively incorporate formal verification methods when: (1) safety requirements are expressible in tractable temporal logics, (2) computational resources permit either real-time verification or pre-computation of safe action sets, (3) environmental uncertainty is bounded and characterizable, and (4) human oversight remains available for edge cases exceeding verification coverage.

References

A. Benton, Eugen Solowjow, and Prithvi Akella. "Verifiable Learned Behaviors via Motion Primitive Composition: Applications to Scooping of Granular Media." *IEEE International Conference on Robotics and Automation*, 2023.

A. Khan, Michael Andrev, M. Murtaza, Sergio Aguilera, Rui Zhang, Jie Ding, Seth Hutchinson, and Ali Anwar. "Safety Aware Task Planning via Large Language Models in Robotics." *arXiv.org*, 2025.

Abdulrahman Althobaiti, Angel Ayala, JingYing Gao, Ali Almutairi, Mohammad Deghat, Imran Razzak, and Francisco Cruz. "How Can LLMs and Knowledge Graphs Contribute to Robot Safety? A Few-Shot Learning Approach." *arXiv.org*, 2024.

Ahmad Hafez, Alireza Naderi Akhormeh, Amr Hegazy, and Amr Alanwar. "Safe LLM-Controlled Robots with Formal Guarantees via Reachability Analysis." *arXiv.org*, 2025.

Aishan Liu, Zonghao Ying, Le Wang, Junjie Mu, Jinyang Guo, Jiakai Wang, Yuqing Ma, et al. "AGENTSAFE: Benchmarking the Safety of Embodied Agents on Hazardous Instructions." *arXiv.org*, 2025.

Akkamahadevi Hanni, Andrew Boateng, and Yu Zhang. "Safe Explicable Robot Planning." *arXiv.org*, 2023.

Aladin Djuhera, Amin Seffo, Masataro Asai, and Holger Boche. ""Don't Do That!": Guiding Embodied Systems Through Large Language Model-Based Constraint Generation." *arXiv.org*, 2025.

Amir Bayat, Alessandro Abate, Necmiye Ozay, and Raphaël M. Jungers. "LLM-Enhanced Symbolic Control for Safety-Critical Applications." *arXiv.org*, 2025.

Ana Davila, Jacinto Colan, and Yasuhisa Hasegawa. "Affordance-Based Disambiguation of Surgical Instructions for Collaborative Robot-Assisted Surgery." *arXiv.org*, 2025.

———. "LLM-Based Ambiguity Detection in Natural Language Instructions for Collaborative Surgical Robots." *arXiv.org*, 2025.

Behrad Rabiei, Mahesh Kumar A.R., Zhirui Dai, Surya L.S.R. Pilla, Qiyue Dong, and Nikolay Atanasov. "LTLCodeGen: Code Generation of Syntactically Correct Temporal Logic for Robot Task Planning." *arXiv.org*, 2025.

Benedict Quartey, Eric Rosen, Stefanie Tellex, and G. Konidaris. "Verifiably Following Complex Robot Instructions with Foundation Models." *IEEE International Conference on Robotics and Automation*, 2024.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Josef Dai, Yuanpei Chen, and Yaodong Yang. "SafeVLA: Towards Safety Alignment of Vision-Language-Action Model via Safe Reinforcement Learning." *arXiv.org*, 2025.

Chen Ding, Cheng Li, Guiling Wu, Liang Qian, Ruifeng Liu, and Xinhao Cai. “SC-RAG-CoT: An Optimization Method for Robotic Instruction Generation.” *2025 International Conference on Mechatronics, Robotics, and Artificial Intelligence (MRAI)*, 2025.

Dan BW Choe, Sundhar Vinodh Sangeetha, Steven Emanuel, Chih-Yuan Chiu, Samuel Coogan, and Shreyas Kousik. “Seeing, Saying, Solving: An LLM-to-TL Framework for Cooperative Robots.” *arXiv.org*, 2025.

Daniel Ekpo, Mara Levy, Saksham Suri, Chuong Huynh, and Abhinav Shrivastava. “VeriGraph: Scene Graphs for Execution Verifiable Robot Planning.” *arXiv.org*, 2024.

Devesh Nath, Haoran Yin, and Glen Chou. “Formal Safety Verification and Refinement for Generative Motion Planners via Certified Local Stabilization.” *arXiv.org*, 2025.

E. Kaigom. “Natural Robot Guidance Using Transformers.” *IEEE International Conference on Emerging Technologies and Factory Automation*, 2023.

Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Changgan Yin, Minghui Li, Lulu Xue, et al. “BadRobot: Jailbreaking Embodied LLMs in the Physical World,” 2024.

Haoyu Wang, Christopher M. Poskitt, Jun Sun, and Jiali Wei. “Pro2Guard: Proactive Runtime Enforcement of LLM Agent Safety via Probabilistic Model Checking.” *arXiv.org*, 2025.

Ike Obi, Vishnunandan L. N. Venkatesh, Weizheng Wang, Ruiqi Wang, Dayoon Suh, T. I. Amosa, Wonse Jo, and Byung-Cheol Min. “SafePlan: Leveraging Formal Logic and Chain-of-Thought Reasoning for Enhanced Safety in LLM-Based Robotic Task Planning.” *arXiv.org*, 2025.

J. Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. “Lang2LTL: Translating Natural Language Commands to Temporal Robot Task Specification.” *arXiv.org*, 2023.

J. Rosser, Jacob Arkin, Siddharth Patki, and T. Howard. “Resolving Ambiguity via Dialogue to Correct Unsynthesizable Controllers for Free-Flying Robots.” *IEEE Aerospace Conference*, 2023.

J. S. Park, Biao Jia, Mohit Bansal, and Dinesh Manocha. “Generating Realtime Motion Plans from Attribute-Based Natural Language Instructions Using Dynamic Constraint Mapping,” 2017.

———. “Generating Realtime Motion Plans from Complex Natural Language Commands Using Dynamic Grounding Graphs.” *arXiv.org*, 2017.

Jason Liu, Ziyi Yang, Benjamin Schornstein, Sam Liang, Ifrah Idrees, Stefanie Tellex, and Ankit Shah. “Lang2LTL: Translating Natural Language Commands to Temporal Specification with Large Language Models,” 2022.

Jeremy Siburian, Keisuke Shirai, C. C. Beltran-Hernandez, Masashi Hamaya, Michael Görner, and Atsushi Hashimoto. “Grounded Vision-Language Interpreter for Integrated Task and Motion Planning.” *arXiv.org*, 2025.

Jiabao Ji, Yongchao Chen, Yang Zhang, R. Kompella, Chuchu Fan, Gaowen Liu, and Shiyu Chang. “Collision- and Reachability-Aware Multi-Robot Control with Grounded LLM Planners.” *arXiv.org*, 2025.

Jiayi Pan, Glen Chou, and D. Berenson. “Data-Efficient Learning of Natural Language to Linear Temporal Logic Translators for Robot Task Specification.” *IEEE International Conference on Robotics and Automation*, 2023.

Jun Wang, David Smith Sundarsingh, Jyotirmoy V. Deshmukh, and Y. Kantaros. “ConformalNL2LTL: Translating Natural Language Instructions into Temporal Logic Formulas with Conformal Correctness Guarantees.” *arXiv.org*, 2025.

Jun Wang, Guocheng He, and Y. Kantaros. “Probabilistically Correct Language-Based Multi-Robot Planning Using Conformal Prediction.” *IEEE Robotics and Automation Letters*, 2024.

———. “Safe Task Planning for Language-Instructed Multi-Robot Systems Using Conformal Prediction.” *arXiv.org*, 2024.

Junhui Huang, Yuhe Gong, Changsheng Li, Xingguang Duan, and L. Figueredo. “ZLATTE: A Geometry-Aware, Learning-Free Framework for Language-Driven Trajectory Reshaping in Human-Robot Interaction,” 2025.

Junle Li, Meiqi Tian, and Bingzhuo Zhong. “Automatic Generation of Safety-Compliant Linear Temporal Logic via Large Language Model: A Self-Supervised Framework.” *arXiv.org*, 2025.

Kaiqu Liang, Zixu Zhang, and J. F. Fisac. “Introspective Planning: Aligning Robots' Uncertainty with Inherent Task

Ambiguity.” *Neural Information Processing Systems*, 2024.
———. “Introspective Planning: Guiding Language-Enabled Agents to Refine Their Own Uncertainty.” *arXiv.org*, 2024.
Kumar Manas, and A. Paschke. “Semantic Role Assisted Natural Language Rule Formalization for Intelligent Vehicle.” *RuleML+RR*, 2023.
Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. “CoT-TL: Low-Resource Temporal Knowledge Representation of Planning Instructions Using Chain-of-Thought Reasoning.” *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2024.
———. “TR2MTL: LLM Based Framework for Metric Temporal Logic Formalization of Traffic Rules.” *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024.
L. Guan, Yifan Zhou, Denis Liu, Yantian Zha, H. B. Amor, and Subbarao Kambhampati. “”Task Success” Is Not Enough: Investigating the Use of Video-Language Models as Behavior Critics for Catching Undesirable Agent Behaviors.” *arXiv.org*, 2024.
Leonardo Santos, Zirui Li, Lasse Peters, Somil Bansal, and Andrea V. Bajcsy. “Updating Robot Safety Representations Online From Natural Language Feedback.” *IEEE International Conference on Robotics and Automation*, 2024.
Lukas Brunke, Yanni Zhang, Ralf Romer, Jack Naimer, Nikola Staykov, Siqi Zhou, and Angela P. Schoellig. “Semantically Safe Robot Manipulation: From Semantic Scene Understanding to Motion Safeguards.” *IEEE Robotics and Automation Letters*, 2024.
Mani Amani, and Reza Akhavian. “Digital Twin-Guided Robot Path Planning: A Beta-Bernoulli Fusion with Large Language Model as a Sensor.” *arXiv.org*, 2025.
Marta Skreta, N. Yoshikawa, Sebastian Arellano-Rubach, Zhi Ji, L. B. Kristensen, Kourosh Darvish, Alán Aspuru-Guzik, F. Shkurti, and Animesh Garg. “Errors Are Useful Prompts: Instruction Guided Task Programming with Verifier-Assisted Iterative Prompting.” *arXiv.org*, 2023.
Maximilian Tolle, Theo Gruner, Daniel Palenicek, Tim Schneider, Jonas Gunster, Joe Watson, Davide Tateo, Puze Liu, and Jan Peters. “Towards Safe Robot Foundation Models Using Inductive Biases.” *arXiv.org*, 2025.
Milan Ganai, Rohan Sinha, Christopher Agia, Daniel Morton, and Marco Pavone. “Real-Time Out-of-Distribution Failure Prevention via Multi-Modal Reasoning.” *arXiv.org*, 2025.
Nishanth Kumar, Fabio Ramos, Dieter Fox, and Caelan Reed Garrett. “Open-World Task and Motion Planning via Vision-Language Model Inferred Constraints.” *arXiv.org*, 2024.
P. Sermanet, Anirudha Majumdar, Alex Irpan, Dmitry Kalashnikov, and Vikas Sindhwani. “Generating Robot Constitutions & Benchmarks for Semantic Safety.” *arXiv.org*, 2025.
Parv Kapoor, Abigail Hammer, Ashish Kapoor, Karen Leung, and Eunsuk Kang. “Pretrained Embeddings as a Behavior Specification Mechanism.” *arXiv.org*, 2025.
Parv Kapoor, Akila Ganlath, Michael Clifford, Changliu Liu, Sebastian Scherer, and Eunsuk Kang. “Constrained Decoding for Robotics Foundation Models.” *arXiv.org*, 2025.
Parv Kapoor, Sai H. Vemprala, and Ashish Kapoor. “Logically Constrained Robotics Transformers for Enhanced Perception-Action Planning.” *arXiv.org*, 2024.
Qian Meng, Jin Peng Zhou, Kilian Q. Weinberger, and H. Kress-Gazit. “INPROVF: Leveraging Large Language Models to Repair High-Level Robot Controllers from Assumption Violations.” *arXiv.org*, 2025.
S. Zhan, Yao Liu, Philip Wang, Zinan Wang, Qineng Wang, Zhian Ruan, Xiangyu Shi, et al. “SENTINEL: A Multi-Level Formal Framework for Safety Evaluation of LLM-Based Embodied Agents,” 2025.
Sabit Hassan, Hye-Young Chung, Xiang Zhi Tan, and Malihe Alikhani. “Coherence-Driven Multimodal Safety Dialogue with Active Learning for Embodied Agents.” *Adaptive Agents and Multi-Agent Systems*, 2024.
Sara Mohammadinejad, Jesse Thomason, and Jyotirmoy V. Deshmukh. “Interactive Learning from Natural Language and Demonstrations Using Signal Temporal Logic.” *arXiv.org*, 2022.
Sathwik Karnik, Zhang-Wei Hong, Nishant Abhangi, Yen-Chen Lin, Tsun-Hsuan Wang, and Pulkit Agrawal. “Em-

bodied Red Teaming for Auditing Robotic Foundation Models." *arXiv.org*, 2024.

Seif Ismail, Antonio Arbues, Ryan Cotterell, René Zurbrügg, and Carmen Amo Alonso. "NARRATE: Versatile Language Architecture for Optimal Control in Robotics." *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2024.

Shaojun Xu, Xusheng Luo, Yutong Huang, Letian Leng, Ruixuan Liu, and Changliu Liu. "Nl2Hltl2Plan: Scaling Up Natural Language Understanding for Multi-Robots Through Hierarchical Temporal Logic Task Specifications." *IEEE Robotics and Automation Letters*, 2024.

Teun van de Laar, Zengjie Zhang, Shuhao Qi, S. Haesaert, and Zhiyong Sun. "VernaCopter: Disambiguated Natural-Language-Driven Robot via Formal Specifications." *arXiv.org*, 2024.

Tsung-Yen Yang, Michael Y Hu, Yinlam Chow, P. Ramadge, and Karthik Narasimhan. "Safe Reinforcement Learning with Natural Language Constraints." *Neural Information Processing Systems*, 2020.

Vanya Cohen, J. Liu, Raymond Mooney, Stefanie Tellex, and David Watkins. "A Survey of Robotic Language Grounding: Tradeoffs Between Symbols and Embeddings." *International Joint Conference on Artificial Intelligence*, 2024.

Vasileios Manginas, Nikolaos Manginas, Edward Stevinson, Sherwin Varghese, Nikos Katzouris, Georgios Paliouras, and Alessio Lomuscio. "A Scalable Approach to Probabilistic Neuro-Symbolic Verification." *arXiv.org*, 2025.

Wanjing Huang, Tongjie Pan, and Yalan Ye. "Graphormer-Guided Task Planning: Beyond Static Rules with LLM Safety Perception." *arXiv.org*, 2025.

William English, Dominic Simon, Rickard Ewetz, and Sumit Kumar Jha. "NSP: A Neuro-Symbolic Natural Language Navigational Planner." *International Conference on Machine Learning and Applications*, 2024.

William Xie, Enora Rice, and N. Correll. "On the Dual-Use Dilemma in Physical Reasoning and Force." *arXiv.org*, 2025.

Yi Wu, Zikang Xiong, Yiran Hu, Shreyash S. Iyengar, Nan Jiang, Aniket Bera, Lin Tan, and Suresh Jagannathan. "SELP: Generating Safe and Efficient Task Plans for Robot Agents with Large Language Models." *IEEE International Conference on Robotics and Automation*, 2024.

Yilin Wu, Ran Tian, Gokul Swamy, and Andrea Bajcsy. "From Foresight to Forethought: VLM-In-the-Loop Policy Steering via Latent Alignment." *Robotics*, 2025.

Yong Qi, Gabriel Kyebambo, Siyuan Xie, Wei Shen, Shenghui Wang, Bitao Xie, Bin He, Zhipeng Wang, and Shuo Jiang. "Safety Control of Service Robots with LLMs and Embodied Knowledge Graphs." *arXiv.org*, 2024.

Yue Meng, Sai H. Vemprala, Rogerio Bonatti, Chuchu Fan, and Ashish Kapoor. "ConBaT: Control Barrier Transformer for Safe Policy Learning." *arXiv.org*, 2023.

Yunhao Yang, Cyrus Neary, and U. Topcu. "Multimodal Pretrained Models for Verifiable Sequential Decision-Making: Planning, Grounding, and Perception." *Adaptive Agents and Multi-Agent Systems*, 2023.

Yunhao Yang, Junyuan Hong, Gabriel J. Perin, Zhiwen Fan, Li Yin, Zhangyang Wang, and U. Topcu. "AD-VF: LLM-Automatic Differentiation Enables Fine-Tuning-Free Robot Planning from Formal Methods Feedback." *arXiv.org*, 2025.

Yunhao Yang, N. Bhatt, Tyler Ingebrand, William Ward, Steven Carr, Zhangyang Wang, and U. Topcu. "Fine-Tuning Language Models Using Formal Methods Feedback." *Conference on Machine Learning and Systems*, 2023.

Yunhao Yang, William Ward, Zichao Hu, Joydeep Biswas, and U. Topcu. "Joint Verification and Refinement of Language Models for Safety-Constrained Planning." *arXiv.org*, 2024.

Zachary Ravichandran, Alexander Robey, Vijay Kumar, George J. Pappas, and Hamed Hassani. "Safety Guardrails for LLM-Enabled Robots." *arXiv.org*, 2025.

Zeyu Feng, Hao Luan, Pranav Goyal, and Harold Soh. "LTLDoG: Satisfying Temporally-Extended Symbolic Constraints for Safe Diffusion-Based Planning." *IEEE Robotics and Automation Letters*, 2024.

Zhendong Chen, ZhanShang Nie, Shixing Wan, JunYi Li, YongTian Cheng, and Shuai Zhao. "An LLM-Powered Natural-to-Robotic Language Translation Framework with Correctness Guarantees." *arXiv.org*, 2025.

Ziming Wang, Qingchen Liu, Jiahu Qin, and Man Li. "Ensuring Safety in LLM-Driven Robotics: A Cross-Layer

Sequence Supervision Mechanism.” *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2024.

Zirui Song, Guangxian Ouyang, Mingzhe Li, Yuheng Ji, Chenxi Wang, Zixiang Xu, Zeyu Zhang, et al. “ManipLVM-R1: Reinforcement Learning for Reasoning in Embodied Manipulation with Large Vision-Language Models.” *arXiv.org*, 2025.

Zitong Bo, Yue Hu, Jinming Ma, Mingliang Zhou, Junhui Yin, Yachen Kang, Yuqi Liu, Tong Wu, Diyun Xiang, and Hao Chen. “Reinforced Embodied Planning with Verifiable Reward for Real-World Robotic Manipulation,” 2025.

Ziyi Yang, S. S. Raman, Ankit Shah, and Stefanie Tellex. “Plug in the Safety Chip: Enforcing Constraints for LLM-Driven Robot Agents.” *IEEE International Conference on Robotics and Automation*, 2023.