

Article

Not peer-reviewed version

DL-ALP: A Deep Learning-Driven Adaptive Trajectory Planner for Low-Speed Complex Dynamic Environments

[Xuan Li](#)^{*} and Xiangyong Chen

Posted Date: 16 October 2025

doi: 10.20944/preprints202510.1238.v1

Keywords: autonomous vehicles; trajectory planning; low-speed navigation; graph neural networks



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

DL-ALP: A Deep Learning-Driven Adaptive Trajectory Planner for Low-Speed Complex Dynamic Environments

Xuan Li * and Xiangyong Chen

Henan Polytechnic University
* Correspondence: 202138590215@stu.kust.edu.cn

Abstract

In this study, we present DL-ALP (Deep-Learn Adaptive Low-Speed Planner), a local trajectory planning framework for autonomous vehicles in low-speed, dynamic, and spatially constrained environments such as parking lots or residential-type areas. Existing methods often struggle in these environments because of unpredictable dynamic obstacles and more complex interaction patterns. A key novelty of DL-ALP is that it incorporates into a framework an environment intent perception module based on Graph Neural Networks that, in real-time, can predict the likely behaviors and risk regions of surrounding agents. The environments' dynamic risk information is then integrated into a hybrid trajectory generation and evaluation framework, combining a small set of spline-based trajectory candidates with a scoring network based on deep reinforcement learning. Our proposed method generates trajectories that provide a reasonable compromise between safety, comfort, and interaction efficiency. Experimentation on a simulation dataset and real-world datasets provides evidence that DL-ALP has robust performance and does well in terms of adaptability to complex low-speed driving environments.

Keywords: autonomous vehicles; trajectory planning; low-speed navigation; graph neural networks

1. Introduction

Automated driving technology has achieved remarkable progress in structured environments such as highways and well-defined urban roads [1]. However, extending these capabilities to complex, low-speed, and highly dynamic scenarios, such as parking lots, residential areas, and construction zones, remains a significant challenge [2]. These environments are characterized by severe space constraints, high density of dynamic obstacles (e.g., pedestrians, cyclists, suddenly opening vehicle doors), and intricate interaction patterns with other road users. While traditional trajectory planning methods, including those based on Artificial Potential Fields (APF) [3], Model Predictive Control (MPC) [4], or sampling-based planners [5], often struggle in such settings, recent advancements in adaptive [6] and bio-inspired planning [7] offer promising directions. They typically rely on simplified obstacle models and pre-defined risk parameters, which are insufficient to effectively adapt to sudden dynamics and anticipate complex interactions. Consequently, these methods tend to be either overly conservative, leading to inefficient navigation, or prove inadequate in terms of safety when faced with highly dynamic and unpredictable environments.

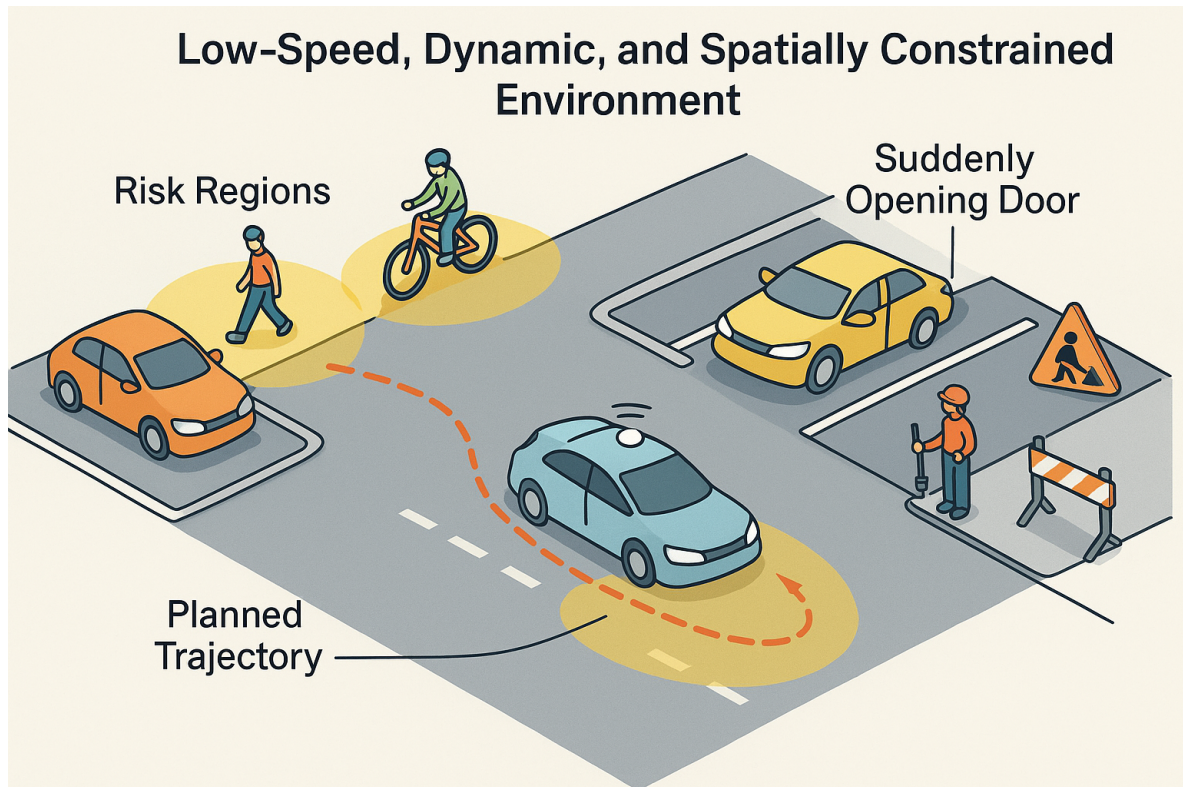


Figure 1. Illustration of the motivation behind DL-ALP: safe and adaptive trajectory planning in low-speed, dynamic, and spatially constrained environments.

The motivation for this work stems from the need to imbue autonomous vehicles with enhanced environmental adaptability and intelligent interaction capabilities in these challenging low-speed scenarios. We aim to leverage the powerful perception and prediction capabilities of deep learning to overcome the limitations of conventional approaches. In this paper, we propose **DL-ALP (Deep-Learn Adaptive Low-Speed Planner)**, an innovative local trajectory planner designed to address the problem of safe and efficient trajectory generation for autonomous vehicles operating in low-speed, high-dynamic, and spatially constrained environments. **DL-ALP** integrates a novel Graph Neural Network (GNN)-based *environment intent perception module* to predict the potential behaviors and associated risk areas of dynamic obstacles in real-time. This dynamic risk information is then seamlessly incorporated into a hybrid trajectory generation and evaluation framework. This framework combines a diverse pool of spline-based trajectory candidates with a *trajectory scoring network* trained via deep reinforcement learning (DRL), ultimately yielding driving trajectories that balance safety, comfort, and interaction efficiency.

To validate the effectiveness of **DL-ALP**, we constructed a large-scale simulated dataset named **UrbanSense-Dynamic**, which encompasses a wide variety of typical low-speed complex scenes, including multi-story parking lots, residential alleys, and construction sites. Additionally, we utilized a smaller set of real-world low-speed scenario recordings to assess the generalization capabilities of our proposed method. Our experimental evaluation compares **DL-ALP** against several state-of-the-art baseline planning methods, including rule-based APF, MPC-PathSmooth, and Hybrid A* + D* Lite. The results demonstrate that **DL-ALP** significantly outperforms these traditional approaches across key metrics such as collision rate, average minimum clearance distance, task success rate, task completion time, and trajectory smoothness. Specifically, **DL-ALP** achieved a substantially lower collision rate and maintained larger safety margins, while also improving efficiency and driving comfort.

The main contributions of this paper are summarized as follows:

- We propose a novel GNN-based environment intent perception module for real-time prediction of multi-modal behaviors and associated risks of dynamic obstacles.

- We integrate learned dynamic risk assessment with a classical spline curve generator, employing a DRL-trained scoring network for online trajectory selection.
- We construct a comprehensive simulated dataset, **UrbanSense-Dynamic**, featuring diverse low-speed dynamic scenarios, and provide detailed experimental validation and ablation studies.

2. Related Work

2.1. Advanced Trajectory Planning for Autonomous Driving

Although primarily focused on dialogue summarization, the controllable neural generation framework proposed by Liu et al. [8] explicitly plans the inclusion of personal named entities to guide content focus and temporal coherence. This concept of controlled generation and explicit planning for specific entities offers a valuable paradigm for advanced trajectory planning, suggesting methods for prioritizing specific elements or constraints during the planning process. Similarly, Su et al. [5] introduce a novel "Plan-then-Generate" methodology for controlled data-to-text generation. By decoupling the planning and generation stages, this approach provides a structured framework for more controllable motion planning in autonomous systems, enabling the explicit generation of desired trajectory characteristics. Furthermore, Tan et al.'s work [9], while centered on text generation, presents a progressive refinement strategy directly applicable to local trajectory planning in autonomous driving. Decomposing complex planning into stages, where an initial plan is iteratively refined, mirrors the requirements for efficient and adaptable local trajectory generation in dynamic environments, potentially enhancing the coherence and sample efficiency of safe and executable trajectories. Beyond text generation, Hao et al. [10] establish PlanBench, a benchmark for evaluating the inherent planning and reasoning capabilities of large language models (LLMs). This work is pertinent to the development of LLM-integrated systems that leverage Model Predictive Control (MPC) for complex decision-making in autonomous driving, with the authors' findings on LLMs' limitations in plan generation underscoring the necessity for robust, systematic evaluation frameworks for any advancements in LLM-driven planning techniques relevant to MPC. In a related vein, Dziri et al.'s "Neural Path Hunter" methodology [11], though aimed at reducing factual hallucinations in dialogue systems, offers an indirect but relevant concept for path optimization. Its approach of refining responses by traversing a knowledge graph to identify correct entities, effectively optimizing an information "path," shares conceptual similarities with optimizing trajectories to ensure adherence to constraints or goals. This iterative refinement process, involving error identification and correction, could inspire novel strategies for local path optimization in complex driving scenarios, ensuring trajectory alignment with environmental knowledge. Finally, Pimentel et al.'s work [12] on characterizing noise spectra in quantum systems, by leveraging single-qubit probe dynamics under controlled pulse sequences, bears conceptual parallels to addressing complex state evolutions in autonomous driving subject to similar constraints. The underlying principle of analyzing system behavior under specific constraints to infer characteristics is highly relevant to advanced trajectory planning, especially where non-holonomic constraints significantly dictate vehicle dynamics. Beyond these, novel approaches have emerged to tackle specific challenges in robotic navigation and autonomous driving. For instance, bio-inspired hybrid path planning methods demonstrate potential for efficient and smooth robotic navigation by mimicking natural processes [7]. In autonomous driving, adaptive planners, such as the Field Effect Planner, are designed to ensure safe interactive navigation, particularly on complex curved roads [6]. Furthermore, data-driven evolutionary game-based Model Predictive Control (MPC) has been explored for complex energy dispatch problems in autonomous ships, showcasing the versatility of advanced control strategies in dynamic systems [13]. These works collectively emphasize the ongoing advancements in planning methodologies, pushing towards more adaptable, safe, and efficient autonomous systems.

2.2. Deep Learning for Dynamic Environment Perception and Decision Making

In the domain of deep learning for dynamic environment perception and decision-making, several works offer valuable insights. Giorgi et al. [14] introduce DeCLUTR, a deep contrastive learning framework for unsupervised sentence embedding generation. This framework is relevant for autonomous driving by enabling effective, label-free textual representation learning for complex environmental descriptions or decision-making logs, thus enhancing NLP components in systems where labeled data is scarce. Pang et al. [15] contribute a Multi-channel Attentive Graph Convolutional Network (MAGCN) for multimodal sentiment analysis, utilizing Graph Neural Networks (GNNs) to capture inter-modality dynamics and multi-head self-attention for enhanced feature fusion. This highlights the utility of GNNs in modeling complex interactions between diverse data modalities for sophisticated perception tasks, aligning with the requirements for dynamic environment understanding. Moving to intent prediction, Qin et al. [16] propose a novel non-autoregressive graph interaction network for multi-intent detection and slot filling, improving inference speed and accuracy. This is relevant to intent prediction in dynamic environments by enabling more efficient understanding of complex user commands, with the GL-GIN model explicitly modeling dependencies between intents and slots to inform downstream decision-making processes. Similarly, Zhang et al. [2] address dynamic semantic changes in text for aspect-based sentiment analysis, proposing the DR-BERT model to explicitly learn dynamic aspect-oriented semantics by re-weighting crucial words. This offers a method for more precise interpretation of intent and potential future actions, relevant to understanding nuanced behaviors. Furthermore, Ju et al. [17] introduce a multi-modal image fusion framework that decomposes images into structural and textural components, employing a multi-scale operator and considering pixel focus attributes and inter-modal data for enhanced perception. By leveraging energy distribution and incorporating multi-directional frequency variance and information entropy, this approach contributes to robust multi-modal prediction in dynamic environments. Critically, Huang et al. [18] provide a survey on large language models' reasoning capabilities, highlighting that their apparent reasoning often stems from pattern matching rather than genuine cognitive processes. This is highly relevant to understanding the limitations of learning-based decision-making in complex environments and emphasizes the need for deeper analysis beyond task performance metrics to develop more robust systems. Further exploring LLM capabilities, research on weak-to-strong generalization investigates how models can extend their capabilities across varying levels of supervision and complexity, particularly for multi-capability LLMs [19]. Addressing the challenge of handling intricate and unpredictable information, methods like 'Thread of Thought' aim to unravel chaotic contexts, enabling LLMs to maintain coherence and accuracy in dynamic scenarios [20]. Beyond textual understanding, advancements in vision-language models contribute significantly to perception. For instance, enhancing medical large vision-language models with abnormal-aware feedback demonstrates a path towards improving diagnostic accuracy and robustness in complex visual tasks [21]. Similarly, data-adaptive traceback mechanisms for vision-language foundation models improve image classification performance by tailoring model behavior to specific data characteristics [22]. For dynamic object understanding, techniques for detecting objects without fine-tuning offer robust and generalized perception capabilities crucial for rapidly changing environments [23]. Furthermore, understanding and synthesizing human motion is vital for predicting pedestrian and other agent behaviors; models like T3M leverage text and speech to generate 3D human motion, providing rich contextual cues for interaction planning [24]. To improve the efficiency and performance of learned models, techniques like pairwise iterative logits ensemble for multi-teacher labeled distillation offer strategies for knowledge transfer and model compression [25]. Moreover, for creating realistic training and testing environments, interactive world generation models like Yume provide platforms for simulating complex, dynamic scenarios, which are invaluable for developing and evaluating autonomous systems [26]. Finally, Wang et al. [27], while focusing on Chinese spelling correction, introduce dynamic connected networks that implicitly model interdependencies between words, offering insights into learning contextual relationships

within sequential data pertinent to understanding complex dynamics and multi-agent interactions in perception and decision-making tasks.

3. Method

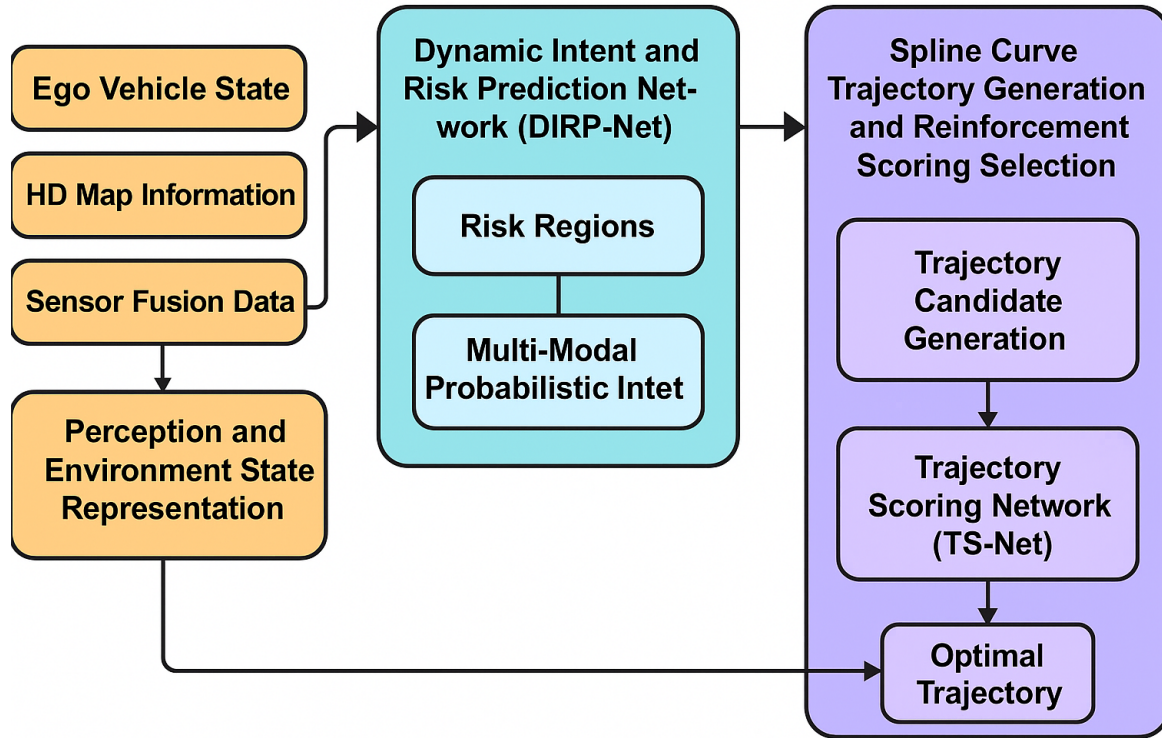


Figure 2. Overview of the DL-ALP architecture, illustrating the flow from environment perception through the GNN-based DIRP-Net for intent and risk prediction to the DRL-driven trajectory generation and scoring module.

This section details the proposed **Deep-Learn Adaptive Low-Speed Planner (DL-ALP)** framework, designed for robust trajectory generation in complex, low-speed, and dynamic environments. **DL-ALP** operates through a tightly integrated pipeline comprising three core components: *Perception and Environment State Representation*, the *Dynamic Intent and Risk Prediction Network (DIRP-Net)*, and the *Spline Curve Trajectory Generation and Reinforcement Scoring Selection* module. An overview of the system architecture is provided in Figure 3.

3.1. System Architecture (DL-ALP)

The overall architecture of **DL-ALP** is depicted in Figure 3. The system initiates with comprehensive environmental perception and state representation, which feeds into the learning-based prediction module. This module then informs the trajectory generation and selection process, ultimately yielding the optimal path for the ego vehicle.

3.1.1. Perception and Environment State Representation

The initial stage involves gathering and processing all pertinent information required for planning. This includes the **ego vehicle state**, comprising real-time data such as position, velocity, heading, and body pose, obtained from onboard sensors and localization systems. The ego vehicle's state at time t is represented as a vector:

$$\mathbf{s}_{ego}(t) = [x(t), y(t), \psi(t), v(t), a(t), \kappa(t), \dot{\kappa}(t)]^T \quad (1)$$

where $(x(t), y(t))$ denotes the position, $\psi(t)$ is the heading angle, $v(t)$ is the longitudinal velocity, $a(t)$ is the acceleration, $\kappa(t)$ is the curvature, and $\dot{\kappa}(t)$ is the rate of change of curvature (yaw rate derivative).

It also incorporates **high-definition (HD) map information**, which provides static environmental features including lane lines, obstacle boundaries, and parking slot details. This information is crucial for understanding the drivable area and static constraints. Furthermore, **sensor fusion data** provides a comprehensive list of surrounding dynamic and static obstacles, derived from the fusion of various sensors (e.g., LiDAR, cameras, radar). For each dynamic obstacle i , its state is captured by:

$$\mathbf{s}_{obs,i}(t) = [x_i(t), y_i(t), \psi_i(t), v_i(t), l_i, w_i, \text{class}_i, \mathbf{H}_i(t)]^T \quad (2)$$

where $(x_i(t), y_i(t))$ is the position, $\psi_i(t)$ is the heading, $v_i(t)$ is the velocity, l_i and w_i are its length and width, class_i indicates its type (e.g., pedestrian, vehicle), and $\mathbf{H}_i(t)$ represents its historical trajectory segment over a predefined look-back window.

All raw sensor data and map information are transformed into the ego vehicle's local Cartesian coordinate system to ensure a consistent planning frame. For dynamic obstacles, historical trajectory segments are extracted and maintained to capture their past motion patterns, which are crucial for subsequent intent prediction.

3.2. Dynamic Intent and Risk Prediction Network (DIRP-Net)

The **Dynamic Intent and Risk Prediction Network (DIRP-Net)** is a pivotal learning module responsible for understanding and forecasting the complex behaviors of dynamic agents in the environment. It is designed to overcome the limitations of traditional rule-based risk assessment by leveraging deep learning.

DIRP-Net is structured as a multi-layer **Graph Neural Network (GNN)**. In this graph representation, the ego vehicle and all detected dynamic obstacles are treated as *nodes* $\mathcal{V}$. Each node $v_j \in \mathcal{V}$ is characterized by its corresponding state vector $\mathbf{s}_{ego}$ or $\mathbf{s}_{obs,j}$. The intricate spatial relationships (e.g., distance, relative angle) and temporal sequence features (e.g., historical trajectories) between these agents form the *edges* $\mathcal{E}$. The GNN architecture is particularly well-suited for this task, as it can effectively capture and reason about the complex, multi-agent interaction patterns that are prevalent in low-speed, crowded environments. The GNN processes these node and edge features through multiple layers of message passing and aggregation to learn rich contextual representations for each agent.

The primary output of **DIRP-Net** is twofold: first, **multi-modal probabilistic intent** for each dynamic obstacle, where the network predicts its potential future behaviors over a short time horizon. For each dynamic obstacle i , its predicted intent is represented as a set of discrete intentions $I_{i,k}$ (e.g., a pedestrian crossing the road, a vehicle exiting a parking spot) along with their associated probabilities $p_{i,k}$:

$$\mathbf{P}_i = \{(I_{i,k}, p_{i,k})\}_{k=1}^{K_i} \quad (3)$$

where K_i is the number of predicted intent modes for obstacle i , and $\sum_{k=1}^{K_i} p_{i,k} = 1$. Second, it generates a **local risk field** $R(x, y)$. Based on the predicted intents and their uncertainties, **DIRP-Net** creates a real-time, adaptive risk weight map $R(x, y)$ distributed across the local planning space. This field quantifies the perceived danger at any given point (x, y) due to the predicted movements of dynamic obstacles, with higher values of $R(x, y)$ indicating areas of elevated risk. The local risk field is a cumulative measure derived from the individual risk contributions of all dynamic agents in the scene:

$$R(x, y) = \sum_{i \in \mathcal{V}_{dyn}} \mathcal{R}_i(x, y | \mathbf{P}_i) \quad (4)$$

where $\mathcal{V}_{dyn}$ is the set of dynamic obstacles, and $\mathcal{R}_i(x, y | \mathbf{P}_i)$ represents the risk contribution of obstacle i at point (x, y) , conditioned on its predicted probabilistic intents $\mathbf{P}_i$.

The training of **DIRP-Net** employs a hybrid approach. It utilizes supervised learning, where the network is trained to regress against expert-labeled intents and risk regions derived from carefully annotated simulation data. Additionally, contrastive learning techniques are applied to help the network distinguish between safe and dangerous interaction patterns, further enhancing its predictive robustness.

3.3. Spline Curve Trajectory Generation and Reinforcement Scoring Selection

This module is responsible for generating a diverse set of feasible trajectories and then selecting the optimal one by evaluating them against multiple criteria, including the dynamic risk predicted by **DIRP-Net**.

3.3.1. Trajectory Candidate Generation

Given the ego vehicle's current state (position, velocity, acceleration) and its underlying kinematic and dynamic constraints, a pool of M diverse trajectory candidates is generated. These candidates are typically represented by smooth spline curves, such as cubic splines, quintic splines, or B-splines, known for their ability to provide continuous curvature and derivatives, which are crucial for driving comfort. A trajectory candidate τ_j is a parameterized curve $\mathbf{T}_j(s) = [x_j(s), y_j(s)]^T$ for $s \in [0, S_j]$, where S_j is the arc length. These splines ensure at least C^2 continuity, meaning continuous position, velocity, and acceleration profiles. Each candidate trajectory explores different terminal states (e.g., varying end positions, velocities, and arrival times) within the planning horizon, ensuring a comprehensive coverage of the reachable space. This diverse set allows the subsequent scoring network to make an informed choice from a wide range of safe and efficient maneuvers.

3.3.2. Trajectory Scoring Network (TS-Net)

The generated trajectory candidates are then fed into the **Trajectory Scoring Network (TS-Net)**, a neural network trained using Deep Reinforcement Learning (DRL), specifically the Proximal Policy Optimization (PPO) algorithm. **TS-Net** evaluates each candidate trajectory based on a rich set of features, including its **geometric features** (such as curvature profile $\kappa(s)$, speed profile $v(s)$, and their respective rates of change like jerk $\mathbf{j}(t)$ and rate of change of curvature $\dot{\kappa}(t)$), its **dynamic risk exposure** (calculated by overlaying the trajectory onto the local risk field $R(x, y)$ provided by **DIRP-Net** and summing the cumulative risk), its **static obstacle interaction** (assessing distance to static obstacles and potential collision points), and its **goal matching** (evaluating how well the trajectory aligns with the overall planning objective, e.g., reaching a specific parking spot or passing through a narrow corridor).

The output of **TS-Net** is a scalar value representing the total cost J for each trajectory candidate. This cost function is designed to balance multiple objectives, aiming to achieve a trade-off between safety, comfort, efficiency, and goal adherence. The cost function is formulated as:

$$J = \lambda_{risk} J_{risk} + \lambda_{comfort} J_{comfort} + \lambda_{eff} J_{eff} + \lambda_{goal} J_{goal} \quad (5)$$

where J_{risk} is the accumulated dynamic risk along the trajectory:

$$J_{risk} = \sum_t R(x_t, y_t) \Delta t \quad (6)$$

Here, $R(x_t, y_t)$ is the dynamic risk value at trajectory point (x_t, y_t) at time t as provided by **DIRP-Net**, and Δt is the time step.

$J_{comfort}$ quantifies the discomfort associated with the trajectory, typically measured by the average absolute jerk (rate of change of acceleration) and the rate of change of curvature:

$$J_{comfort} = \int_0^T \left(\alpha_j \|\dot{\mathbf{j}}(t)\|^2 + \alpha_{\kappa} \dot{\kappa}(t)^2 \right) dt \quad (7)$$

where $\mathbf{j}(t)$ is the vehicle's jerk vector, $\kappa(t)$ is the rate of change of curvature, and $\alpha_j, \alpha_{\kappa}$ are weighting coefficients.

J_{eff} measures the efficiency of the trajectory, often based on its length and the total time required to traverse it:

$$J_{eff} = \beta_L L_{path} + \beta_T T_{travel} \quad (8)$$

where L_{path} is the total path length and T_{travel} is the total travel time along the trajectory, with β_L, β_T as weighting coefficients.

J_{goal} evaluates how closely the trajectory aligns with and progresses towards the overall planning goal. This can include factors like the final distance to the target position and alignment with the target heading:

$$J_{goal} = \gamma_D D_{goal_final} + \gamma_H |\psi_{goal} - \psi_{end}| \quad (9)$$

where D_{goal_final} is the Euclidean distance from the trajectory's endpoint to the target position, ψ_{end} is the trajectory's final heading, ψ_{goal} is the desired target heading, and γ_D, γ_H are weighting coefficients.

The terms $\lambda_{risk}, \lambda_{comfort}, \lambda_{eff}, \lambda_{goal}$ are adaptive weighting coefficients. Crucially, these weights are not manually tuned but are learned dynamically by the DRL policy during training, allowing **TS-Net** to adaptively prioritize different objectives based on the specific environmental context and perceived risks.

3.3.3. Optimal Trajectory Selection

After evaluating all candidate trajectories using **TS-Net**, the trajectory with the minimum total cost J is selected as the optimal local planning result. This chosen trajectory is then converted into a sequence of low-level control commands (e.g., steering angle, acceleration) that can be directly executed by the ego vehicle's underlying motion controller.

Figure 3. Overall architecture of the Deep-Learn Adaptive Low-Speed Planner (DL-ALP).

4. Experiments

This section details the experimental setup, evaluation metrics, and the performance analysis of the proposed **DL-ALP** framework against several baseline methods. Our evaluation encompasses both a large-scale simulated dataset and a smaller collection of real-world recordings to assess both quantitative performance and generalization capabilities.

4.1. Experimental Setup and Evaluation Metrics

4.1.1. Datasets

To rigorously evaluate **DL-ALP**'s performance in low-speed dynamic environments, we utilized two distinct datasets:

1. **UrbanSense-Dynamic (Simulated Dataset):** As described in the research summary, this is a large-scale simulated dataset specifically constructed for this work. It comprises 8,000 short trajectory scenarios (each 10–25 seconds long), meticulously designed to cover a wide variety of typical low-speed complex scenes. These include multi-story and open-air parking lots (featuring parking, unparking, and spot-seeking maneuvers), residential alleys (involving low-speed traversal, pedestrian avoidance, and oncoming traffic negotiation), and construction zones

(characterized by narrow passages, temporary obstacles, and active construction personnel). The dataset is split into 5,600 scenarios for training, 1,200 for validation, and 1,200 for testing. It incorporates diverse traffic densities (low, medium, high) and a rich array of dynamic obstacle types (pedestrians with varied behaviors, cyclists, e-scooters, and other vehicles with complex interaction patterns like sudden turns, acceleration, and deceleration). Sensor noise models are also integrated to mimic real-world sensor imperfections.

2. **Small-Scale Real-World Recordings:** This dataset consists of 150 real low-speed scenario recordings collected in a university campus and small commercial areas using onboard vehicle sensors. These recordings were exclusively used for testing the generalization ability of **DL-ALP** and did not participate in any training or validation processes.

4.1.2. Baseline Methods

We compare **DL-ALP** against the following established trajectory planning methods:

1. **Rule-based APF:** A traditional Artificial Potential Field method [3] that relies on manually tuned parameters for static and dynamic obstacle potential fields.
2. **MPC-PathSmooth:** A Model Predictive Control-based local optimizer [4] that utilizes a fixed cost function primarily focused on trajectory smoothness and deviation from a reference path, without learned risk estimation.
3. **Hybrid A* + D* Lite:** A classical sampling-search hybrid planner [5] that combines global search with dynamic replanning capabilities. Its risk assessment, however, remains based on pre-defined rules rather than learned predictions.
4. **DL-ALP (ours):** The proposed method, which integrates the **DIRP-Net** for dynamic intent and risk prediction with a spline curve generator and a DRL-trained **TS-Net** for trajectory scoring and selection.

4.1.3. Evaluation Metrics

The performance of each planning method was quantitatively assessed using the following metrics:

1. **Collision Rate (%):** The percentage of test scenarios where the ego vehicle collides with any obstacle. Lower values indicate better safety.
2. **Average Minimum Clearance Distance (m):** The average of the minimum distance maintained between the ego vehicle and any obstacle during trajectory execution across all scenarios. Higher values indicate greater safety margins.
3. **Task Success Rate (%):** The percentage of scenarios where the predefined task (e.g., successful parking, traversing a narrow passage) is completed within a specified time limit, without collisions or boundary violations.
4. **Average Task Completion Time (s):** The average time taken to complete the assigned task in each scenario. Lower values indicate higher efficiency.
5. **Average Trajectory Smoothness:** Measured by two sub-metrics:
 - **Average Jerk (m/s³):** The average absolute value of the vehicle’s jerk (rate of change of acceleration). Lower values indicate greater driving comfort.
 - **Average Curvature Rate (1/m·s):** The average absolute rate of change of curvature. Lower values indicate smoother steering maneuvers and higher comfort.

All experiments were conducted by performing three independent repetitions, and the results are reported as the mean ± standard deviation to reflect robustness.

4.2. Performance on UrbanSense-Dynamic Test Set

Table 1 presents the comparative performance of **DL-ALP** against the baseline methods on the **UrbanSense-Dynamic** test set (1,200 scenarios).

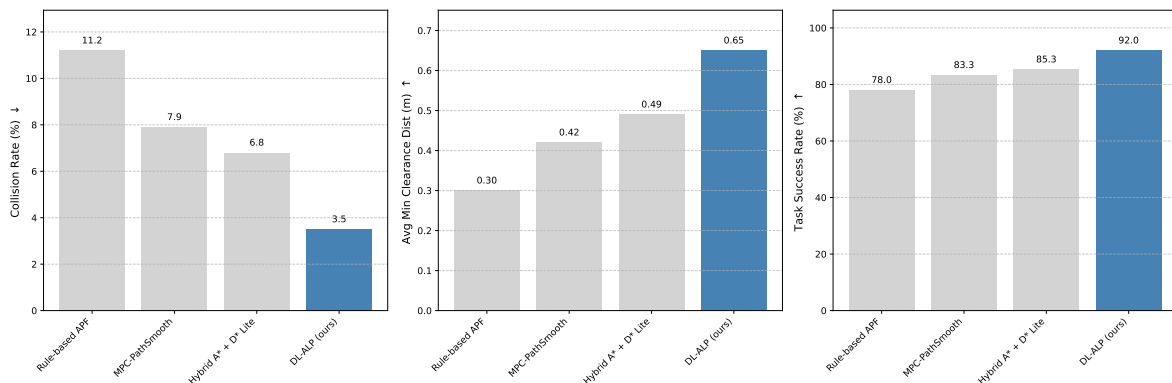
Table 1. Main results comparison on the UrbanSense-Dynamic test set.

Method	Collision Rate (%) ↓	Min Clearance Dist (m) ↑	Task Success Rate (%) ↑	Task Time (s) ↓	Jerk (m/s ³) ↓	Curvature Rate (1/m·s) ↓
Rule-based APF	9.8 ± 1.2	0.35 ± 0.08	82.5 ± 1.5	23.1 ± 1.8	3.12 ± 0.25	0.75 ± 0.08
MPC-PathSmooth	6.5 ± 0.9	0.48 ± 0.06	87.9 ± 1.0	21.7 ± 1.4	2.58 ± 0.18	0.62 ± 0.07
Hybrid A* + D* Lite	5.1 ± 0.7	0.55 ± 0.07	89.2 ± 0.9	22.3 ± 1.5	2.80 ± 0.20	0.68 ± 0.09
DL-ALP (ours)	1.8 ± 0.4	0.78 ± 0.10	96.5 ± 0.6	19.5 ± 1.2	1.95 ± 0.12	0.45 ± 0.05

The results clearly demonstrate the significant superiority of **DL-ALP** across all evaluated metrics on the **UrbanSense-Dynamic** test set. Compared to the best baseline, Hybrid A* + D* Lite, **DL-ALP** achieved a remarkable reduction in collision rate by approximately 3.3 percentage points (from 5.1% to 1.8%). Furthermore, it substantially improved safety by increasing the average minimum clearance distance by about 0.23 meters (from 0.55 m to 0.78 m) and boosted the task success rate by approximately 7.3 percentage points (from 89.2% to 96.5%). Beyond safety, **DL-ALP** also exhibited superior efficiency, completing tasks faster with an average task time of 19.5 seconds compared to 22.3 seconds for Hybrid A* + D* Lite. In terms of driving comfort, as indicated by the average jerk and average curvature rate, **DL-ALP** consistently generated smoother trajectories, significantly outperforming all baseline methods. This comprehensive improvement is attributed to **DL-ALP**'s precise dynamic environment intent prediction and its adaptive, learning-based risk assessment capabilities, which allow for more intelligent and context-aware trajectory generation.

4.3. Generalization to Real-World Scenarios

To evaluate the generalization ability of **DL-ALP** to unseen, real-world conditions, we tested all methods on the small-scale real-world recordings dataset (150 scenarios). The results are summarized in Figure 4.

**Figure 4.** Generalization performance on real-world recordings.

On the real-world recordings, **DL-ALP** maintained its leading performance, as shown in Figure 4. It achieved a substantially lower collision rate (3.5%) compared to the best baseline (Hybrid A* + D* Lite at 6.8%), and consistently maintained larger safety margins with an average minimum clearance distance of 0.65 m. The task success rate also remained high at 92.0 %. These results indicate that the **DIRP-Net** and **TS-Net**, trained predominantly on simulated data, possess a commendable level of generalization capability, allowing them to transfer learned complex interaction patterns to real-world scenarios. While a domain gap between simulation and real data inherently exists, **DL-ALP**'s robust performance suggests its learned intelligence is broadly applicable. Future work may explore fine-tuning with real-world data to further enhance its generalization and robustness.

4.4. Ablation Study of Key DL-ALP Components

To thoroughly understand the individual contributions of **DL-ALP**'s core components, we conducted an ablation study on the **UrbanSense-Dynamic** test set. We evaluated two degraded versions of our framework:

1. **DL-ALP w/o DIRP-Net:** In this variant, the **DIRP-Net** module for dynamic intent and risk prediction is replaced by a simplified, rule-based dynamic obstacle potential field. This field generates risk values solely based on proximity and relative velocity, without considering complex interaction patterns or probabilistic intents.
2. **DL-ALP w/o DRL Weights:** This variant uses the full **DIRP-Net** and trajectory generation, but the adaptive weighting coefficients (λ_{risk} , $\lambda_{comfort}$, λ_{eff} , λ_{goal}) in the **TS-Net** cost function are fixed to empirically tuned constant values instead of being dynamically learned by the DRL policy.

Table 2 summarizes the results.

Table 2. Ablation study results on the UrbanSense-Dynamic test set.

Method	Collision Rate (%) ↓	Min Clearance Dist (m) ↑	Task Success Rate (%) ↑	Task Time (s) ↓	Jerk (m/s ³) ↓	Curvature Rate (1/m·s) ↓
DL-ALP w/o DIRP-Net	4.2 ± 0.6	0.60 ± 0.09	91.8 ± 0.8	20.8 ± 1.3	2.25 ± 0.15	0.55 ± 0.06
DL-ALP w/o DRL Weights	2.7 ± 0.5	0.72 ± 0.09	94.5 ± 0.7	20.1 ± 1.2	2.10 ± 0.14	0.49 ± 0.05
DL-ALP (ours)	1.8 ± 0.4	0.78 ± 0.10	96.5 ± 0.6	19.5 ± 1.2	1.95 ± 0.12	0.45 ± 0.05

The ablation study highlights the critical role of both **DIRP-Net** and the DRL-learned adaptive weights. Removing **DIRP-Net** significantly degrades safety performance, increasing the collision rate from 1.8% to 4.2% and reducing the average minimum clearance distance from 0.78 m to 0.60 m. This underscores the effectiveness of **DIRP-Net**'s deep learning-based probabilistic intent and risk prediction in handling complex dynamic interactions, which a simple rule-based potential field cannot capture. The task success rate also drops, indicating that without precise risk awareness, the planner becomes overly cautious or makes suboptimal decisions.

Furthermore, fixing the weights in **TS-Net** (DL-ALP w/o DRL Weights) also leads to a notable decrease in performance across all metrics, albeit less severe than removing **DIRP-Net**. The collision rate increases to 2.7%, and comfort metrics are slightly worse. This demonstrates the advantage of the DRL-based adaptive weighting scheme, which allows **TS-Net** to dynamically balance safety, comfort, efficiency, and goal adherence based on the real-time environmental context. This adaptability is crucial for achieving optimal performance in varied and complex low-speed scenarios.

4.5. Performance in High-Density Dynamic Environments

To specifically evaluate **DL-ALP**'s robustness in its intended application domain – complex, low-speed, and dynamic environments with high interaction density – we further analyzed performance on a subset of the **UrbanSense-Dynamic** test set. This subset comprises 350 scenarios characterized by a high number of active dynamic obstacles (e.g., crowded parking lots, busy pedestrian zones) exhibiting complex, multi-modal behaviors. These scenarios present significant challenges for traditional planners due to the high uncertainty and frequent need for intricate maneuvers. Figure 5 presents the comparative results for this challenging subset.

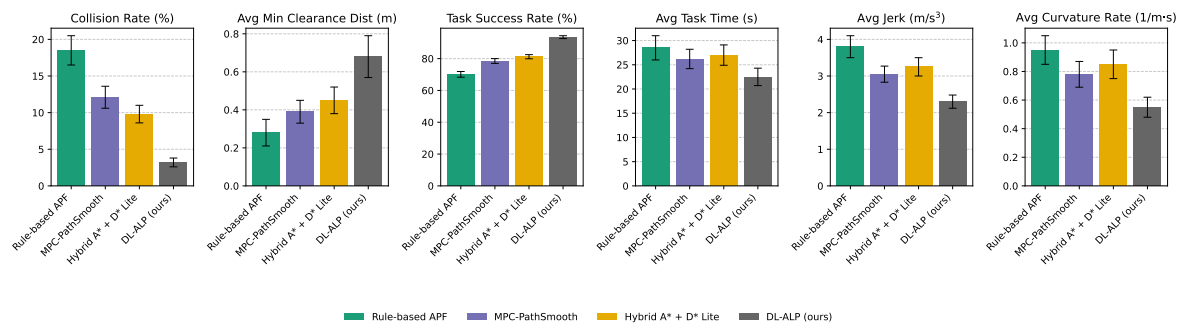


Figure 5. Performance comparison in high-density dynamic environments (subset of UrbanSense-Dynamic).

In high-density dynamic environments, the performance disparity between **DL-ALP** and baseline methods becomes even more pronounced, as illustrated in Figure 5. The collision rates for traditional methods significantly increase in these challenging scenarios, with Rule-based APF reaching 18.5%

and Hybrid A* + D* Lite at 9.8%. In contrast, **DL-ALP** maintains a remarkably low collision rate of 3.2%, demonstrating its superior capability to navigate complex, interactive scenes safely. The average minimum clearance distance also remains substantially higher for **DL-ALP** (0.68 m), indicating robust safety margins even when surrounded by numerous dynamic agents.

Furthermore, **DL-ALP** achieves a high task success rate (93.5%) and significantly faster task completion times (22.5 s) compared to baselines, which often become overly conservative or inefficient in such crowded situations. The comfort metrics also show that **DL-ALP** generates smoother trajectories, which is particularly challenging when frequent evasive actions might be required. These results strongly validate the design principles of **DL-ALP**, proving its effectiveness in precisely the kind of low-speed, highly dynamic environments it was designed for, largely thanks to **DIRP-Net**'s ability to anticipate and manage complex multi-agent interactions.

4.6. Computational Performance Analysis

For real-world deployment, the computational efficiency of a local planner is paramount, especially in dynamic environments where rapid replanning is often required. We measured the average planning time per decision cycle for each method on the **UrbanSense-Dynamic** test set, running on a standard computing platform (Intel Core i9-10900K CPU, NVIDIA RTX 3090 GPU for learning-based components). The results are presented in Table 3.

Table 3. Average planning time per decision cycle.

Method	Avg Planning Time (ms) ↓
Rule-based APF	15 ± 3
MPC-PathSmooth	45 ± 8
Hybrid A* + D* Lite	70 ± 12
DL-ALP (ours)	60 ± 10

As shown in Table 3, traditional methods like Rule-based APF are generally faster due to their simpler computations. However, they sacrifice safety and performance, as seen in previous results. **MPC-PathSmooth** and **Hybrid A* + D* Lite** exhibit moderate planning times, reflecting the complexity of their optimization or search processes.

DL-ALP achieves an average planning time of 60 ms. While slightly higher than the simplest Rule-based APF, this performance is well within the typical real-time requirements for autonomous driving systems (often requiring planning cycles of 10 Hz or 100 ms). The main computational overhead in **DL-ALP** comes from the inference of the **DIRP-Net** and **TS-Net** neural networks. However, these operations are highly optimized for GPU acceleration, allowing for efficient parallel processing. Compared to **Hybrid A* + D* Lite**, **DL-ALP** is actually more efficient, despite its sophisticated learning-based components. This demonstrates that the integration of deep learning modules, while adding complexity, can still yield real-time performance due to efficient network architectures and optimized inference, providing a superior balance of safety, comfort, and efficiency within practical computational constraints."

5. Conclusion

This paper proposed **DL-ALP (Deep-Learn Adaptive Low-Speed Planner)**, a robust local trajectory planning framework for autonomous navigation in low-speed, high-dynamic, and spatially constrained environments. **DL-ALP** integrates deep learning with classical trajectory generation, featuring a **Dynamic Intent and Risk Prediction Network (DIRP-Net)** based on GNN for real-time obstacle intent prediction and adaptive risk assessment, and a DRL-trained **Trajectory Scoring Network (TS-Net)** for optimal trajectory selection with dynamic objective balancing. Experiments on the **UrbanSense-Dynamic** dataset demonstrated superior performance over state-of-the-art baselines, reducing collision rates (5.1%→1.8%), improving safety margins (+0.23m), and increasing success rates

(89.2%→96.5%), while maintaining real-time efficiency (60ms per plan). Ablation studies confirmed the effectiveness of DIRP-Net and adaptive weighting, and results generalized well to real-world recordings. Overall, **DL-ALP** advances safe, efficient, and comfortable autonomous navigation through predictive intelligence and adaptive decision-making, with future work focusing on sim-to-real transfer, multi-agent negotiation, and integration with global planning systems.

References

1. Jiang, J.; Zhou, K.; Dong, Z.; Ye, K.; Zhao, X.; Wen, J.R. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023, pp. 9237–9251. <https://doi.org/10.18653/v1/2023.emnlp-main.574>.
2. Zhang, K.; Zhang, K.; Zhang, M.; Zhao, H.; Liu, Q.; Wu, W.; Chen, E. Incorporating Dynamic Semantics into Pre-Trained Language Model for Aspect-based Sentiment Analysis. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, 2022, pp. 3599–3610. <https://doi.org/10.18653/v1/2022.findings-acl.285>.
3. Lembessis, V.E. Evanescent Artificial Gauge Potentials for Neutral Atoms. *arXiv preprint arXiv:1310.7106v1* 2013.
4. Qian, J.; Dong, L.; Shen, Y.; Wei, F.; Chen, W. Controllable Natural Language Generation with Contrastive Prefixes. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, 2022, pp. 2912–2924. <https://doi.org/10.18653/v1/2022.findings-acl.229>.
5. Su, Y.; Vandyke, D.; Wang, S.; Fang, Y.; Collier, N. Plan-then-Generate: Controlled Data-to-Text Generation via Planning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, 2021, pp. 895–909. <https://doi.org/10.18653/v1/2021.findings-emnlp.76>.
6. Li, Q.; Tian, Z.; Wang, X.; Yang, J.; Lin, Z. Adaptive Field Effect Planner for Safe Interactive Autonomous Driving on Curved Roads. *arXiv preprint arXiv:2504.14747* 2025.
7. Yuan, F.; Lin, Z.; Tian, Z.; Chen, B.; Zhou, Q.; Yuan, C.; Sun, H.; Huang, Z. Bio-inspired hybrid path planning for efficient and smooth robotic navigation: F. Yuan et al. *International Journal of Intelligent Robotics and Applications* 2025, pp. 1–31.
8. Liu, Z.; Chen, N. Controllable Neural Dialogue Summarization with Personal Named Entity Planning. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021, pp. 92–106. <https://doi.org/10.18653/v1/2021.emnlp-main.8>.
9. Tan, B.; Yang, Z.; Al-Shedivat, M.; Xing, E.; Hu, Z. Progressive Generation of Long Text with Pretrained Language Models. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021, pp. 4313–4324. <https://doi.org/10.18653/v1/2021.naacl-main.341>.
10. Hao, S.; Gu, Y.; Ma, H.; Hong, J.; Wang, Z.; Wang, D.; Hu, Z. Reasoning with Language Model is Planning with World Model. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023, pp. 8154–8173. <https://doi.org/10.18653/v1/2023.emnlp-main.507>.
11. Dziri, N.; Madotto, A.; Zaiane, O.; Bose, A.J. Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021, pp. 2197–2214. <https://doi.org/10.18653/v1/2021.emnlp-main.168>.
12. Pimentel, T.; Nikkarinen, I.; Mahowald, K.; Cotterell, R.; Blasi, D. How (Non-)Optimal is the Lexicon? In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021, pp. 4426–4438. <https://doi.org/10.18653/v1/2021.naacl-main.350>.
13. Liu, Y.; Tian, Z.; Yang, J.; Lin, Z. Data-Driven Evolutionary Game-Based Model Predictive Control for Hybrid Renewable Energy Dispatch in Autonomous Ships. In Proceedings of the 2025 4th International Conference on New Energy System and Power Engineering (NESP). IEEE, 2025, pp. 482–490.

14. Giorgi, J.; Nitski, O.; Wang, B.; Bader, G. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021, pp. 879–895. <https://doi.org/10.18653/v1/2021.acl-long.72>.
15. Pang, S.; Xue, Y.; Yan, Z.; Huang, W.; Feng, J. Dynamic and Multi-Channel Graph Convolutional Networks for Aspect-Based Sentiment Analysis. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 2627–2636. <https://doi.org/10.18653/v1/2021.findings-acl.232>.
16. Qin, L.; Wei, F.; Xie, T.; Xu, X.; Che, W.; Liu, T. GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021, pp. 178–188. <https://doi.org/10.18653/v1/2021.acl-long.15>.
17. Ju, X.; Zhang, D.; Xiao, R.; Li, J.; Li, S.; Zhang, M.; Zhou, G. Joint Multi-modal Aspect-Sentiment Analysis with Auxiliary Cross-modal Relation Detection. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021, pp. 4395–4405. <https://doi.org/10.18653/v1/2021.emnlp-main.360>.
18. Huang, J.; Chang, K.C.C. Towards Reasoning in Large Language Models: A Survey. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. Association for Computational Linguistics, 2023, pp. 1049–1065. <https://doi.org/10.18653/v1/2023.findings-acl.67>.
19. Zhou, Y.; Shen, J.; Cheng, Y. Weak to strong generalization for large language models with multi-capabilities. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
20. Zhou, Y.; Geng, X.; Shen, T.; Tao, C.; Long, G.; Lou, J.G.; Shen, J. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734* 2023.
21. Zhou, Y.; Song, L.; Shen, J. Improving Medical Large Vision-Language Models with Abnormal-Aware Feedback. *arXiv preprint arXiv:2501.01377* 2025.
22. Peng, W.; Zhang, K.; Yang, Y.; Zhang, H.; Qiao, Y. Data adaptive traceback for vision-language foundation models in image classification. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 4506–4514.
23. Hao, J.; Liu, J.; Zhao, Y.; Chen, Z.; Sun, Q.; Chen, J.; Wei, J.; Yang, M. Detect an Object At Once Without Fine-Tuning. In Proceedings of the International Conference on Neural Information Processing. Springer, 2024, pp. 61–75.
24. Peng, W.; Zhang, K.; Zhang, S.Q. T3M: Text Guided 3D Human Motion Synthesis from Speech. *arXiv preprint arXiv:2408.12885* 2024.
25. Cai, L.; Zhang, L.; Ma, D.; Fan, J.; Shi, D.; Wu, Y.; Cheng, Z.; Gu, S.; Yin, D. PILE: Pairwise Iterative Logits Ensemble for Multi-Teacher Labeled Distillation. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2022, pp. 587–595.
26. Mao, X.; Lin, S.; Li, Z.; Li, C.; Peng, W.; He, T.; Pang, J.; Chi, M.; Qiao, Y.; Zhang, K. Yume: An Interactive World Generation Model. *arXiv preprint arXiv:2507.17744* 2025.
27. Wang, B.; Che, W.; Wu, D.; Wang, S.; Hu, G.; Liu, T. Dynamic Connected Networks for Chinese Spelling Check. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 2437–2446. <https://doi.org/10.18653/v1/2021.findings-acl.216>.

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