KnowVal: A Knowledge-Augmented and Value-Guided Autonomous Driving System

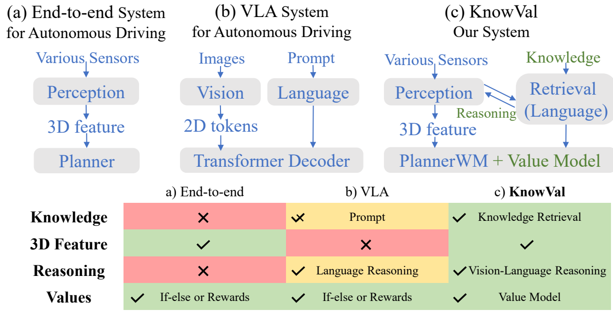
Zhongyu Xia[1] Wenhao Chen[1] Yongtao Wang[1*] Ming-Hsuan Yang[2]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]University of California, Merced
{xiazhongyu,wyt}@pku.edu.cn wenhaochen@stu.pku.edu.cn mhyang@ucmerced.edu

Abstract

Visual–language reasoning, driving knowledge, and value alignment are essential for advanced autonomous driving systems. However, existing approaches largely rely on data-driven learning, making it difficult to capture the complex logic underlying decision-making through imitation or limited reinforcement rewards. To address this, we propose KnowVal, a new autonomous driving system that enables visual–language reasoning through the synergistic integration of open-world perception and knowledge retrieval. Specifically, we construct a comprehensive driving knowledge graph that encodes traffic laws, defensive driving principles, and ethical norms, complemented by an efficient LLM-based retrieval mechanism tailored for driving scenarios. Furthermore, we develop a human-preference dataset and train a Value Model to guide interpretable, value-aligned trajectory assessment. Experimental results show that our method substantially improves planning performance while remaining compatible with existing architectures. Notably, KnowVal achieves the lowest collision rate on nuScenes and state-of-the-art results on Bench2Drive.

1. Introduction

Autonomous driving, a key application of AI, serves as a comprehensive testbed for 3D perception, prediction, reasoning, and planning capabilities. Current state-of-the-art autonomous driving systems mainly follow two paradigms: (a) End-to-end autonomous driving, which leverages 3D feature extraction to directly decode planning trajectories; and (b) Vision–Language–Action (VLA) models, which integrate perception and control through multi-modal large language models. However, an ideal autonomous driving system must operate in open, dynamic, and highly uncertain environments, requiring decisions that are safe, reliable, and explainable while adhering to human social values and norms, which poses significant challenges to existing paradigms.



	a) End-to-end	b) VLA	c) **KnowVal**
Knowledge	✗	✓✗ Prompt	✓ Knowledge Retrieval
3D Feature	✓	✗	✓
Reasoning	✗	✓ Language Reasoning	✓ Vision-Language Reasoning
Values	✓ If-else or Rewards	✓ If-else or Rewards	✓ Value Model

Figure 1. **Comparison of End-to-End, VLA, and our KnowVal.** Our 3D vision system enables visual–language reasoning through the mutual guidance of perception and retrieval, understands *law and morality* via knowledge retrieval, and integrates *values* through a dedicated value model.

The first challenge lies in enabling effective interaction between vision and language to achieve genuine visual–language reasoning. Existing end-to-end autonomous driving models lack language-grounded reasoning capabilities, while VLA models typically confine reasoning to chain-of-thought or purely linguistic processes, without allowing the reasoning outcomes to influence perception. As illustrated in Figure 1, we propose a visual–language reasoning system in which perception and retrieval mutually guide each other: the model retrieves relevant knowledge from open-world perception, and when retrieval identifies missing information, it prompts the perception module to refine its observations accordingly.

The second challenge lies in the absence of an interpretable and structured knowledge base. Existing end-to-end and VLA models primarily rely on data-driven learning, attempting to infer complex decision logic from the limited variability of human behavior—an inherently difficult task. Some approaches introduce handcrafted conditional rules or reinforcement learning rewards, but these manually designed components cover only narrow scenarios and

[*]Corresponding author.

fail to generalize to complex, dynamic environments. In contrast, autonomous driving already has a rich and authoritative source of knowledge: traffic laws and regulations, which provide a comprehensive foundation for safe and lawful driving. Furthermore, abundant online resources, such as instructional driving videos, offer valuable insights into driving techniques and principles. Hence, we fully exploit these resources to construct a knowledge graph of traffic laws, moral principles, and defensive driving behaviors, enabling dynamic knowledge retrieval via perception queries. This allows the model to interpretably reason about traffic laws, moral principles, and defensive driving behaviors.

The third challenge concerns the autonomous driving system's worldview and values. Recent advances highlight the importance of a world model—a module that predicts future world states from current observations and actions—reflecting the predictive and reflective abilities of the autonomous driving systems. However, possessing such a worldview (*i.e.*, world model) alone is insufficient for effective decision-making: after forecasting possible futures, the system must also judge whether those outcomes are desirable, requiring a dedicated value assessment mechanism. Existing paradigms often approximate this ability through data-driven learning or handcrafted rules, both of which limit generalization and interpretability. To address this, we introduce a Value Model grounded in retrieved knowledge and trained on a human-preference dataset, enabling value-aligned and interpretable trajectory evaluation.

To sum up, we propose a novel autonomous driving system in this work, in which perception and knowledge retrieval mutually guide each other, supplying richer contextual information to the decision-making module that integrates both a world model and a value model. The key contributions of this work are:

- We propose a new autonomous driving system, KnowVal, that enables visual–language reasoning through the synergistic interaction between perception and knowledge retrieval.
- We construct a comprehensive driving knowledge graph encompassing traffic laws, defensive driving principles, and moral considerations, and develop an efficient LLM-based retrieval mechanism for it.
- We design a planner that integrates a world model for future-state prediction and a Value Model for outcome evaluation, enabling value-aligned decision-making. A human-preference dataset is also curated to train the Value Model.
- Extensive experiments show that KnowVal maintains strong compatibility with existing methods, and achieves the lowest collision rate on nuScenes and state-of-the-art performance on Bench2Drive.

2. Related Work

Atonomous Driving Systems. The prevailing paradigms for autonomous driving systems can be broadly categorized into three classes: the traditional modular pipeline, the end-to-end approach, and the Vision–Language–Action (VLA) model. Several models have been benchmarked on real-world datasets such as nuScenes [1]. ST-P3 [6] first integrates perception, prediction, and planning into an end-to-end framework. UniAD [7] follows a BEV-based feature extraction strategy similar to BEVFormer [15], with intricate feature propagation across task-specific decoders. VAD [12] adopts a fully vectorized scene representation for trajectory planning, while PARA-Drive [25] further investigates multi-task supervision and module interdependencies within end-to-end architectures. GenAD [32] utilizes BEV representations to obtain sparse detection and mapping features for planning, whereas SparseDrive [23] extracts such features directly from image inputs to mitigate cumulative errors and improve prediction accuracy. Recently, MomAD [22] introduces momentum-aware planning for enhanced temporal consistency, and BridgeAD [31] refines planning results by leveraging historical prediction and trajectory information. Finally, DiffusionDrive [16] employs a truncated diffusion model to generate diverse trajectory candidates, while HENet++ [28] jointly extracts foreground instance and dense panoramic features using attention mechanisms to improve prediction and planning.

With the advancement of 3D physical simulators, many recent works [7, 12, 16] leverage synthetic environments such as CARLA [2] and Bench2Drive [10] for training and evaluation. Despite this progress, these methods still largely rely on imitation learning paradigms. TCP [26] introduces a dual-branch architecture that integrates trajectory and control signals to enhance planning accuracy. ThinkTwice [9] proposes a two-stage planner that first generates candidate trajectories and then refines them based on scene context, partially reflecting the concept of a world model. DriveAdapter [8] utilizes an expert planner for knowledge distillation, improving planning robustness. AD-MLP [30] extracts priors from historical trajectories using multilayer perceptrons, while DriveTransformer [11] adopts parallel queries for sparse interactions with sensor data in streaming multi-task learning. With the emergence of Vision–Language Models (VLMs), several methods have adapted them for driving command generation, among which ORION [4] is representative. SimLingo [19] further aligns natural-language prompts (e.g., "adjust speed," "change lane") with corresponding action sequences, advancing interpretability and language-conditioned control.

However, these approaches lack vision–language reasoning, explicit knowledge utilization, and value-based judgment. Our proposed system overcomes these limitations through perception-guided knowledge retrieval and

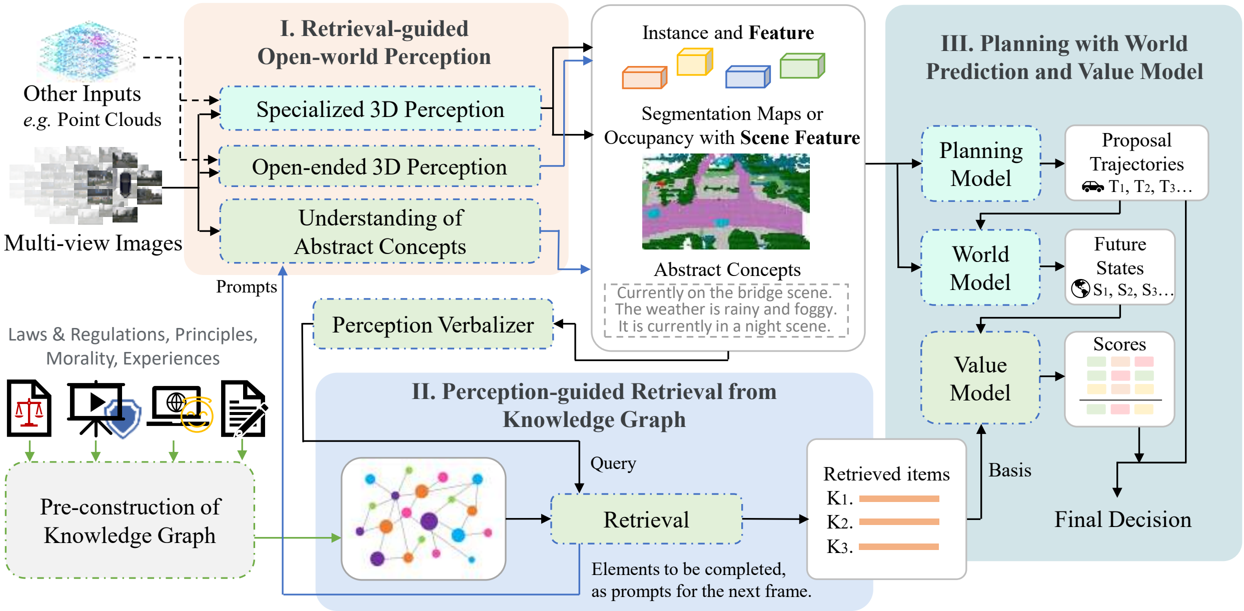

I. Retrieval-guided Open-world Perception
Other Inputs
e.g. Point Clouds
Multi-view Images
Specialized 3D Perception
Open-ended 3D Perception
Understanding of Abstract Concepts
Prompts
Instance and Feature
Segmentation Maps or Occupancy with Scene Feature
Abstract Concepts
Currently on the bridge scene.
The weather is rainy and foggy.
It is currently in a night scene.
Perception Verbalizer
Laws & Regulations, Principles, Morality, Experiences
Pre-construction of Knowledge Graph
II. Perception-guided Retrieval from Knowledge Graph
Query
Retrieval
Retrieved items
K1.
K2.
K3.
Elements to be completed, as prompts for the next frame.
III. Planning with World Prediction and Value Model
Planning Model
Proposal Trajectories
T1, T2, T3…
World Model
Future States
S1, S2, S3…
Value Model
Scores
Basis
Final Decision

Figure 2. **Overall architecture of KnowVal.** KnowVal first performs *retrieval-guided open-world perception* to extract instance-, panoramic-, and concept-level information. It then conducts *perception-guided retrieval* from the knowledge graph to obtain relevant knowledge. Finally, decision-making is achieved through integrated planning, world prediction, and value assessment. KnowVal is compatible with existing methods. The Specialized Perception, Planning Model, and World Model are obtained by modifying and fine-tuning existing methods. Feature propagation across modules ensures comprehensive information flow throughout the system.

knowledge-guided value assessment, enabling more interpretable and value-aligned decision-making.

Retrieval Method. To align systems with complex knowledge principles, integrating external knowledge is essential. Early retrieval-based methods, such as [18], adopt a simple linear pipeline—indexing documents, retrieving the most similar chunks to a query, and generating responses based on them. The introduction of Retrieval-Augmented Generation (RAG) [13] improves this process by incorporating pre-retrieval and post-retrieval stages to enhance contextual relevance and generation quality. Modular RAG [20, 29] extends this idea by decomposing the linear RAG pipeline into a flexible architecture with specialized modules arranged in dynamic configurations. GraphRAG [3] constructs a summarized knowledge graph over text corpora, providing global semantic structure and enabling reasoning beyond vector search. More recently, LightRAG [5] introduces a dual-level retrieval mechanism over a graph-enhanced index that supports incremental updates, allowing efficient and adaptive RAG without expensive re-indexing.

However, the aforementioned retrieval-augmented methods, though effective for language models, cannot be directly applied to autonomous driving systems. They inevitably paraphrase and compress retrieved knowledge, which can introduce hallucinations and distort factual content. Moreover, these approaches are primarily designed for chat-based AI, where retrieval is driven by user-issued textual prompts. To meet the unique demands of autonomous driving, we propose a perception-guided retrieval mechanism that grounds knowledge access in visual perception rather than user input. In addition, we refine the retrieval process to ensure that retrieved entries remain faithful to the original texts, thereby preserving factual accuracy and reliability in downstream reasoning and decision-making.

3. Method

3.1. Reasoning between Perception and Retrieval

As illustrated in Figure 3, KnowVal comprises three main components: (i) retrieval-guided open-world perception, (ii) perception-guided retrieval from the knowledge graph, and (iii) planning with world prediction and value assessment. KnowVal achieves visual–language reasoning through the mutual guidance between perception and retrieval, thereby enriching both perceptual understanding and knowledge utilization. Information from these two modules is subsequently propagated to the planning component as structured representations and feature embeddings, ensuring comprehensive decision-making while maintaining an end-to-end architecture.

Retrieval-guided Open-world Perception. Open-world perception in KnowVal encompasses three key capabilities. The first is Specialized Perception, a standard component in most end-to-end autonomous driving models, which enables the recognition and localization of common semantic categories such as vehicles, pedestrians, and drivable areas. The second is Open-ended 3D Perception, which identifies and localizes long-tail or uncommon ob-

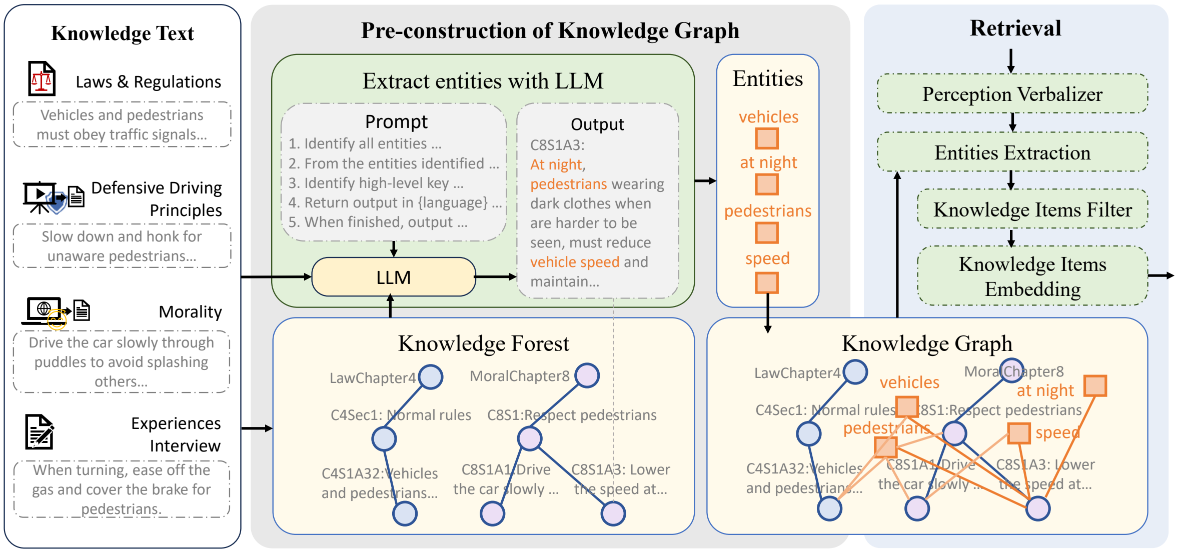

Knowledge Text
Laws & Regulations
Vehicles and pedestrians must obey traffic signals...
Defensive Driving Principles
Slow down and honk for unaware pedestrians...
Morality
Drive the car slowly through puddles to avoid splashing others...
Experiences Interview
When turning, ease off the gas and cover the brake for pedestrians.
Pre-construction of Knowledge Graph
Extract entities with LLM
Prompt
1. Identify all entities ...
2. From the entities identified ...
3. Identify high-level key ...
4. Return output in {language} ...
5. When finished, output ...
Output
C8S1A3:
At night,
pedestrians wearing dark clothes when are harder to be seen, must reduce vehicle speed and maintain...
LLM
Entities
vehicles
at night
pedestrians
speed
Knowledge Forest
LawChapter4
MoralChapter8
C4Sec1: Normal rules
C8S1:Respect pedestrians
C4S1A32:Vehicles and pedestrians...
C8S1A1:Drive the car slowly ...
C8S1A3: Lower the speed at...
Knowledge Graph
LawChapter4
MoralChapter8
vehicles
at night
C4Sec1: Normal rules
C8S1:Respect pedestrians
pedestrians
speed
C4S1A32:Vehicles and pedestrians...
C8S1A1:Drive the car slowly ...
C8S1A3: Lower the speed at...
Retrieval
Perception Verbalizer
Entities Extraction
Knowledge Items Filter
Knowledge Items Embedding

Figure 3. **Pre-construction of the knowledge graph and retrieval process in KNOWVAL.** We collect diverse driving-related resources—including laws, regulations, defensive driving principles, moral guidelines, and experiential knowledge—to construct an initial *knowledge forest* based on textual structures. Large language models (LLMs) are then used to extract entities and define vertices and edges, forming a structured *knowledge graph*. During inference, KNOWVAL generates queries enriched with 3D perception information to retrieve and rank relevant entries from the knowledge graph by descending relevance.

jects (e.g., fire trucks, standing water) without requiring explicit user prompts. This capability is implemented using VL-SAMv2 [17] and OpenAD [27]. The third is Abstract Concept Understanding, which captures scene-level or contextual attributes beyond direct object perception—such as determining whether the environment is a bridge or tunnel, or whether it is daytime or nighttime—achieved via the VLM within VL-SAMv2 [17]. Additionally, this module supports retrieval guidance: the retrieval component from the previous timestep identifies elements requiring further perception, which are then passed to the VLM for targeted processing.

Perception-guided retrieval from knowledge graph. We collect traffic laws, regulations, and driver interview transcripts, and employ multimodal large language models (MLLMs) to extract experiential knowledge, ethical considerations, and defensive driving principles from online video sources. Subsequently, we utilize LLMs to construct a knowledge graph, as described in Section 3.2.1. During inference, a Perception Verbalizer transforms structured perception outputs into query texts, which are then used by an LLM-based retrieval module to obtain knowledge entries ranked in descending order of relevance. Simultaneously, the retrieval module identifies elements requiring additional perception, which are forwarded to the perception component in the next timestep for refinement. The retrieval process is further detailed in Section 3.2.2.

Planning with world prediction and value model. Current planning modules predominantly adopt Transformer or RNN architectures, which inherently possess varying degrees of capability for future-state prediction. Building upon this foundation, we enhance both the planning and world models to generate diverse candidate trajectories and forecast corresponding future states. We then introduce a Value Model that evaluates the desirability of these candidate trajectories and their predicted world states based on retrieved knowledge entries, as detailed in Section 3.3.1. To train this model, we construct a preference dataset comprising extensive trajectory–outcome samples, further described in Section 3.3.2.

规划模型用于生成不同的待选轨迹

价值模型用于结合知识图谱进行评估

3.2. Perception-guided Knowledge Retrieval.

3.2.1. Pre-Construction of Knowledge Graphs

We design a knowledge graph construction method that preserves the complete fidelity of original knowledge clauses while simultaneously revealing latent semantic relationships between these clauses and query entities.

Step 1: Knowledge collection and structured forest construction. We gather information from a variety of sources, including traffic laws and regulations, principles extracted from educational videos on defensive driving, transcripts from interviews with various drivers, and compiled ethical guidelines. For educational videos on defensive driving, we employ Gemini 2.5-Pro with carefully designed prompts to extract key driving principles. The extracted texts are then categorized and organized into chapters according to the driving scenarios and strategies they describe. For driver interview transcripts, we classify the content by driving behaviors and contexts (e.g., left turns, intersections) to produce a structured textual representation. In the

四种不同的知识源

未来状态的预测用到了Transformer或RNN架构。

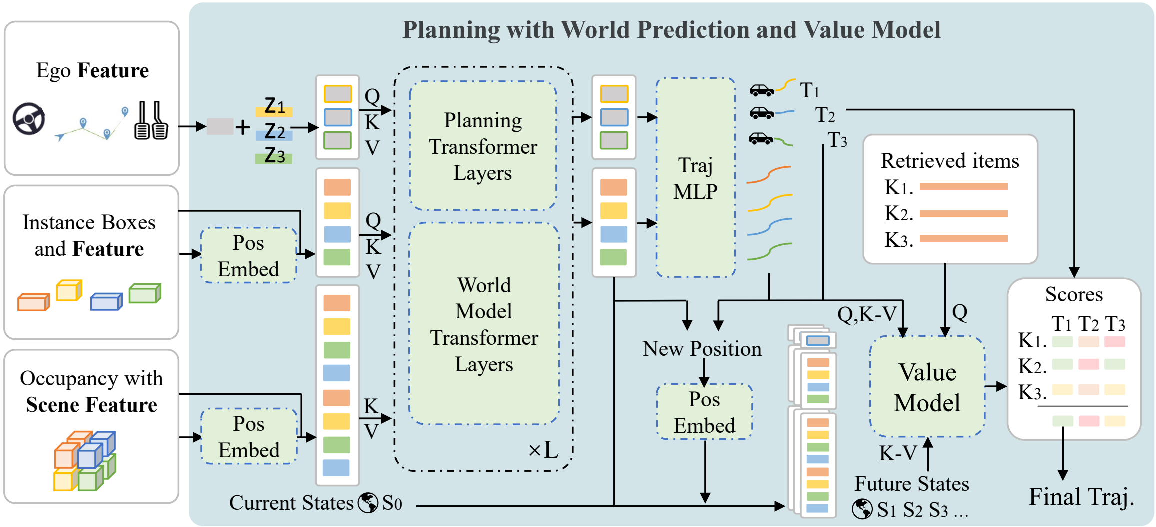


Figure 4. **Planning process of KnowVal.** We design a trajectory generation method compatible with existing Transformer- and RNN-based planners. By combining future world state prediction with value model assessment, KnowVal produces diverse candidate trajectories and makes interpretable final decisions.

case of laws and regulations, their inherent hierarchical organization already exhibits a forest-like structure, which we directly adopt as the foundation for our knowledge forest. This design allows us to directly utilize their intrinsic hierarchical organization to form the knowledge forest. Each knowledge clause node stores the raw, unaltered text, while non-leaf nodes contain the corresponding titles and descriptive information. All nodes in the forest are labeled as *native*. This tree-based organization preserves the original hierarchical structure of the knowledge corpus and serves as the initial form of the knowledge graph.

Step 2: Entity Linking via LLM to Transform the Forest into a Graph. To break down the "knowledge silos" among independent trees, we leverage LLMs to automatically discover and link semantic relationships across the disparate branches of the knowledge forest. We iterate through each knowledge clause leaf node and employ a powerful LLM (e.g., GPT-4 or Qwen) to perform a controlled entity extraction task. Our prompt instructs the model to identify key entities within the clause text corresponding to predefined conceptual categories, including: *Traffic-Sign-Device* (e.g., *traffic signal*, *stop sign*), *Road-User* (e.g., *pedestrian*, *bicycle*), *Driving-Maneuver* (e.g., *turning*, *merging*), and *Road-Condition* (e.g., *wet pavement*, *construction zone*). After entity extraction, we create a new entity node in the knowledge graph for each unique entity and connect it to all corresponding *native* nodes containing that entity. The LLM then assigns semantic weights and relationship descriptions to the edges. Inspired by [5], we further employ an LLM-empowered profiling function to generate text keys for both nodes and edges. Each key is a concise phrase facilitating efficient retrieval, and multiple keys can be generated to encode broader semantic themes across connected entities. This process transforms the initially isolated knowledge forest into a highly interconnected, semantically rich knowledge graph, forming the structural backbone for the online retrieval mechanism described in Section 3.2.2.

3.2.2. Retrieval

We design a retrieval process that ranks results by importance and relevance. This process guarantees retrieval results faithful to the original text, thereby safeguarding against model hallucinations.

Step 1: Perception verbalizer. First, a templated verbalizer converts perception outputs of varying structures into a unified structured natural language. For the 3D instances, the verbalizer directly converts the bounding box and semantic information into textual descriptions. For the semantic map or occupancy results, the verbalizer first uses the Breadth-First Search algorithm to identify semantically connected blocks, handling discontinuous boundaries via a dynamic connectivity threshold. Upon obtaining connected blocks, it also converts their positional and semantic information into textual descriptions, which are then combined with abstract conceptual perception results, navigational data, and user instructions to form the retrieval query.

感知语言化，将第一部分得到的场景语言化便于知识图谱进行检索

Step 2: Entities Extraction. We leverage a lightweight LLM to perform a two-layer keyword extraction from the retrieval query. The first layer extracts macro-level context keywords (e.g., *Driving Security*, *Scene Analysis*), while the second layer extracts specific entity and event keywords (e.g., *Pedestrians*, *Rainy Weather*). Subsequently, these keywords are used to query initial relevant entity nodes and edges within the knowledge graph by the index keys. To capture richer contextual information, we perform a Top-K nearest-neighbor expansion on the graph, incorporating nodes closely connected to the initial set.

Table 1. **Comparison of planning results on *nuScenes*.** L2 denotes the L2 error between the predicted and human driving trajectories.

Method	Input	UniAD Metrics								VAD/STP3 Metrics							
		L2 (m) ↓				Collision Rate (%) ↓				L2 (m) ↓				Collision Rate (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
ST-P3 [6]	C	1.72	3.26	4.86	3.28	0.44	1.08	3.01	1.51	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
OccNet [24]	C	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72	-	-	-	-	-	-	-	-
UniAD [7]	C	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31	0.45	0.70	1.04	0.73	0.62	0.58	0.63	0.61
VAD-Base [12]	C	0.54	1.15	1.98	1.22	0.04	0.39	1.17	0.53	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
PARA-Drive [25]	C	-	-	-	-	-	-	-	-	0.25	0.46	0.74	0.48	0.14	0.23	0.39	0.25
SparseDrive [23]	C	0.44	0.92	1.69	1.01	0.07	0.19	0.71	0.32	0.29	0.58	0.96	0.61	0.01	0.05	0.18	0.08
MomAD [22]	C	0.43	0.88	1.62	0.98	0.06	0.16	0.68	0.30	0.31	0.57	0.91	0.60	0.01	0.05	0.22	0.09
SSR [14]	C	0.24	0.65	1.36	0.75	0.00	0.10	0.36	0.15	0.18	0.36	0.63	0.39	0.01	0.04	0.12	0.06
BridgeAD-B [31]	C	-	-	-	-	-	-	-	-	0.28	0.55	0.92	0.58	0.00	0.04	0.20	0.08
DiffusionDrive [16]	C	-	-	-	-	-	-	-	-	0.27	0.54	0.90	0.57	0.03	0.05	0.16	0.08
GenAD [32]	C	0.36	0.83	1.56	0.91	0.06	0.23	1.00	0.43	0.28	0.49	0.78	0.52	0.08	0.14	0.34	0.19
KnowVal-GenAD	C	0.46	1.35	2.72	1.51	0.00	0.20	0.80	0.33 -0.10	0.28	0.60	1.07	0.65	0.01	0.08	0.25	0.11 -0.08
HENet++ [28]	RC	0.39	1.11	2.36	1.29	0.00	0.06	0.33	0.13	0.24	0.50	0.91	0.55	0.01	0.03	0.10	0.05
KnowVal-HENet++	RC	0.42	1.19	2.35	1.32	0.00	0.10	0.25	**0.12** -0.01	0.26	0.54	0.94	0.58	0.00	0.01	0.08	**0.03** -0.02

Step 3: Knowledge Items Filter and Embedding. From all extracted nodes, we filter for only those explicitly tagged as *native*. These nodes link directly to the original, verbatim knowledge clauses that have not been processed or summarized by any LLM. This filtering step is the fundamental differentiator of our system from conventional RAG, as it completely eliminates the risk of information distortion introduced by LLM processing. The top-N_K relevant filtered items are converted into vector representations in this process, making them suitable for subsequent use in the Value Model. Meanwhile, the retrieval LLM outputs a list of items for supplementary perception, which are then fed back into the subsequent perception stage.

3.3. Planning with World Prediction and Value Model

3.3.1. Planning with World View and Values

As illustrated in Figure 4, we first extend the planner to function as both a planning and world model, generating diverse trajectories and predicting future states. Taking HENet++ [28] as an example, we inject N_T distinct Gaussian noise vectors into the ego-vehicle features, thereby generating N_T ego-specific queries that decode into N_T distinct trajectories. The baseline end-to-end model is fine-tuned with these modifications and an additional diversity constraint: when the L2 distance e between two planned trajectories falls below a predefined threshold τ, the term $(\tau - e)$ is added to the loss function as a penalty. This encourages the planner to explore a wider range of feasible trajectories while maintaining consistency and realism.

Next, for each candidate trajectory $\mathcal{T}i$, KnowVal performs item-wise value assessment based on the retrieved knowledge entries K_j. The Value Model follows a Transformer Encoder–MLP Decoder architecture. For the i-th trajectory and the j-th knowledge entry feature $f_{K_j} \in \mathbb{R}^{N_{\text{tokens}} \times C}$, we take $max(f_{K_j}) \in \mathbb{R}^C$ and concatenate it with the positional encoding of $\mathcal{T}i$ and its ego-vehicle features to form the query $Q_{i,j}$. Through this process, N_K query tokens are generated for each candidate trajectory. These tokens are projected linearly and fed into a Transformer layer, while the future-state tokens S_i—which incorporate ego-vehicle, instance, and occupancy-grid features, each with a positional embedding corresponding to $\mathcal{T}i$—serve as key–value pairs. After L Transformer iterations, the updated N_K features are decoded by the MLP to produce N_K scalar values $s_{i,j} \in [-1, 1]$. A value near -1 indicates behavior inconsistent with knowledge-based reasoning, 1 represents positive compliance, and 0 denotes irrelevance to the rule.

We then apply a weighted-decay scoring strategy to aggregate the per-rule evaluations into a single score for each trajectory. The total value is defined as a normalized weighted average of scores across all retrieved rules, sorted by their relevance:

$$\text{Score}(\mathcal{T}i) = \frac{1}{Z} \sum_{j=1}^{N_K} \gamma^{j-1} \cdot s_{i,j} \tag{1}$$

where $Z = \sum_{j=1}^{N_K} \gamma^{j-1}$ is a normalization constant. This weighting ensures that the most relevant knowledge clauses contribute most significantly to decision-making, enabling interpretable, knowledge-driven trajectory evaluation. Finally, the trajectory with the highest overall score is selected as the final planning output.

3.3.2. Constructing Preference Dataset for Value Model

To train and evaluate the Value Model, we construct the Preference Dataset for Values, which provides quantitative supervision for the abstract notion of whether a driving behavior complies with a specific knowledge clause. The dataset consists of 160K trajectory–knowledge pairs, each accompanied by corresponding scene states (feature vectors and BEV renderings) and ground-truth value annotations. We use 80% of the data for training and reserve the remaining 20% for validation and testing.

Step 1: Future States and Retrieved Basis. Our data construction pipeline begins by employing the retrieval engine described in Section 3.2.2 to obtain a set of relevant knowledge clauses Basis $= K_1, K_2, \ldots$ for each driving scenario in nuScenes [1]. We then collect the corresponding future-state vectors, BEV renderings of perception and prediction results, and other contextual information for each scenario.

Step 2: Annotation Scores for Supervision. We design precise prompts and employ Qwen-VL-Max to automatically generate preliminary compliance scores for each data sample. These scores serve as supervision signals, indicating how well each driving behavior aligns with the corresponding knowledge clauses. To ensure annotation reliability, we conduct a manual review to identify and correct any apparent errors or outliers. During training, the loss function is defined as the mean squared error (MSE) between the predicted value and the annotated score in the constructed dataset.

4. Experiments

4.1. Datasets and Metrics

We evaluate our approach on two widely used autonomous driving benchmarks: the open-loop real-world dataset nuScenes [1] and the closed-loop simulation benchmark Bench2Drive (B2D) [10]. nuScenes includes sensor data from six surround-view cameras, radar, and LiDAR across 1,000 driving scenes. Its planning metrics consist of the L2 error between predicted and human trajectories and the collision rate. Bench2Drive (B2D) provides 10,000 sampled driving segments for training and offers closed-loop evaluation environments with detailed simulator configurations. B2D employs two key metrics: Driving Score and Success Rate. The Driving Score is a composite measure that integrates Route Completion and Infraction Penalty, while the Success Rate reflects the percentage of successfully completed driving tasks.

4.2. Implementation Details

We employ Qwen2.5-3B for retrieval and knowledge embedding, with the number of retrieved knowledge entries set to $N_K = 16$. For the planner, the number of candidate trajectories is $N_T = 20$, and the Value Model performs $L = 3$ iterative reasoning steps. The decay factor used in the total value assessment is $\gamma = 0.7$. To increase trajectory diversity, baseline models on nuScenes are fine-tuned for three epochs using eight A100 GPUs, while models on Bench2Drive are fine-tuned for one epoch under the same configuration. The Value Model is trained separately on eight V100 GPUs for 50 epochs using the AdamW optimizer with a cosine annealing learning rate schedule.

Table 2. **Closed-loop results on Bench2Drive.** DS denotes Driving Score, and SR denotes Success Rate (%).

Method	Expert	Paradigm	DS↑	SR↑
AD-MLP [30]	Think2Drive	E2E	18.05	00.00
TCP [26]	Think2Drive	E2E	40.70	15.00
TCP-traj [26]	Think2Drive	E2E	59.90	30.00
UniAD-Base [7]	Think2Drive	E2E	45.81	16.36
VAD [12]	Think2Drive	E2E	42.35	15.00
ThinkTwice [9]	Think2Drive	E2E	62.44	31.23
DriveAdapter [8]	Think2Drive	E2E	64.22	33.08
DriveTrans. [11]	Think2Drive	E2E	63.46	35.01
ORION [4]	Think2Drive	VLA	77.74	54.62
DiffusionDrive [16]	PDM-Lite	E2E	77.68	52.72
SimLingo [19]	PDM-Lite	VLA	85.07	67.27
KnowVal-SimLingo	PDM-Lite	KnowVal	**88.42** +3.35	**69.03** +1.76

4.3. Main Results on Driving Benchmarks

Open-loop Evaluation on nuScenes. As reported in Table 1, KnowVal achieves the lowest Collision Rate among all existing methods on nuScenes. Although the L2 distance is slightly higher, this primarily indicates deviations from human driving trajectories rather than degraded planning quality. In practice, KnowVal tends to discover safer or more efficient driving strategies that may not be reflected in human demonstrations. These results also confirm that the KnowVal can be seamlessly integrated into existing end-to-end architectures such as GenAD and HENet++, enhancing both safety and generalization.

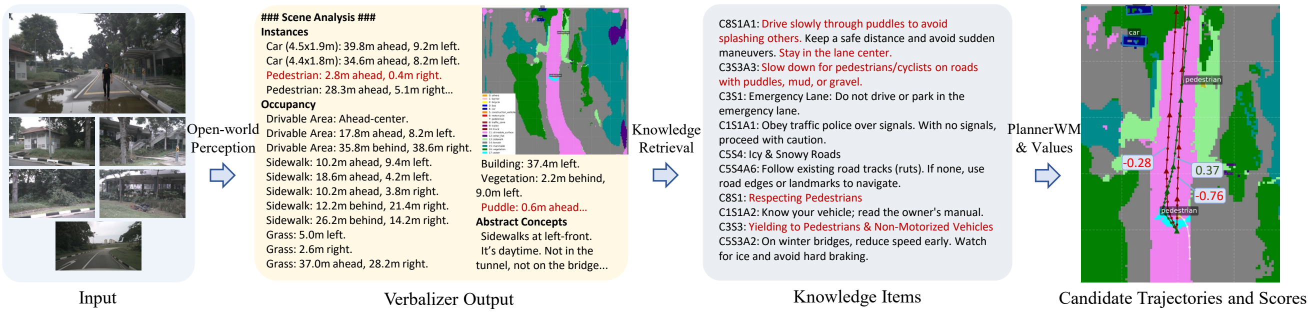
Closed-loop Evaluation on Bench2Drive. As shown in Table 2, KnowVal achieves a further **3.35** improvement in Driving Score and a **1.76%** gain in Success Rate over the VLA-based method SimLingo, establishing new state-of-the-art performance on Bench2Drive. These results demonstrate the effectiveness of our knowledge-grounded planning framework, enabling the autonomous driving system to make safer, more robust decisions in complex and dynamic driving environments.

4.4. Qualitative Analysis

Figure 5 illustrates a representative example of the inference process in our proposed KnowVal, highlighting its reasoning effectiveness. Comparative results with baseline methods are presented in Figure 6. Compared to existing approaches, our paradigm exhibits a superior ability to interpret and act upon ethical and legal considerations during decision-making. It is important to note that current benchmarks lack both sufficient evaluation data and appropriate metrics to assess many forms of improper driving behavior. In the examples shown in Figure 6, actions such as failing to yield to pedestrians or overtaking within tunnels do not affect standard metrics like collision rate or driving score, underscoring the limitations of existing evaluation protocols.

4.5. Ablation

Ablation on Overall System. As shown in Table 3, integrating the knowledge retrieval and value model compo-



Scene Analysis
Instances
Car (4.5x1.9m): 39.8m ahead, 9.2m left.
Car (4.4x1.8m): 34.6m ahead, 8.2m left.
Pedestrian: 2.8m ahead, 0.4m right.
Pedestrian: 28.3m ahead, 5.1m right…
Occupancy
Drivable Area: Ahead-center.
Drivable Area: 17.8m ahead, 8.2m left.
Drivable Area: 35.8m behind, 38.6m right.
Sidewalk: 10.2m ahead, 9.4m left.
Sidewalk: 18.6m ahead, 4.2m left.
Sidewalk: 10.2m ahead, 3.8m right.
Sidewalk: 12.2m behind, 21.4m right.
Sidewalk: 26.2m behind, 14.2m right.
Grass: 5.0m left.
Grass: 2.6m right.
Grass: 37.0m ahead, 28.2m right.
Building: 37.4m left.
Vegetation: 2.2m behind, 9.0m left.
Puddle: 0.6m ahead…
Abstract Concepts
Sidewalks at left-front.
It's daytime. Not in the tunnel, not on the bridge…

C8S1A1: Drive slowly through puddles to avoid splashing others. Keep a safe distance and avoid sudden maneuvers. Stay in the lane center.
C3S3A3: Slow down for pedestrians/cyclists on roads with puddles, mud, or gravel.
C3S1: Emergency Lane: Do not drive or park in the emergency lane.
C1S1A1: Obey traffic police over signals. With no signals, proceed with caution.
C5S4: Icy & Snowy Roads
C5S4A6: Follow existing road tracks (ruts). If none, use road edges or landmarks to navigate.
C8S1: Respecting Pedestrians
C1S1A2: Know your vehicle; read the owner's manual.
C3S3: Yielding to Pedestrians & Non-Motorized Vehicles
C5S3A2: On winter bridges, reduce speed early. Watch for ice and avoid hard braking.

Figure 5. **Illustration of the inference process of KnowVal.** KnowVal perceives long-tail scenarios, retrieves ethical knowledge related to pedestrians and puddles, and employs the Value Model to select appropriate actions.

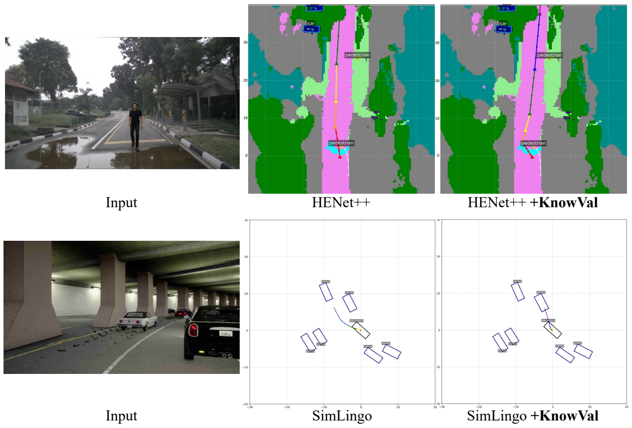


Figure 6. **Qualitative comparison between KnowVal and baseline methods.** The upper example, edited following [21] on a nuScenes scenario, illustrates the **ethical commonsense**: "When passing through accumulated water, slow down to avoid splashing pedestrians." The lower example, sampled from Bench2Drive, demonstrates the **legal regulation**: "Inside tunnels, vehicles must not overtake across solid lines." Unlike baseline methods, KnowVal correctly handles both scenarios.

Table 3. **Ablation studies on the KnowVal system.** The baseline model is GenAD. K&V: using the proposed knowledge retrieval and value model. OW: incorporating open-ended perception and understanding of abstract concepts. RgP: using retrieval results to guide supplementary perception. N_K: the number of retrieved knowledge entries. W_K: applying weighting to the total score. N_T: the number of candidate trajectories.

K&V	OW	RgP	N_K	W_K	N_T	L2(m)↓	Col↓
			16	✓	20	0.91	0.43
✓			16	✓	20	0.89	0.38
✓	✓		16	✓	20	1.47	0.34
✓	✓	✓	16	✓	20	1.51	0.33
✓	✓	✓	16	✓	10	1.56	0.37
✓	✓	✓	16	✓	20	1.51	0.33
✓	✓	✓	16	✓	30	1.52	0.33
✓	✓	✓	8	✓	20	1.29	0.36
✓	✓	✓	16		20	1.67	0.38
✓	✓	✓	16	✓	20	1.51	0.33
✓	✓	✓	24	✓	20	1.49	0.34

Table 4. **Comparison on different Value Model architectures.**

Input S_i	Output	Traning Scale	MSE↓	MAE↓
BEV Image	Total Score	10K	0.35	0.59
BEV Image	Total Score	128K	0.22	0.44
Scene Feature	Total Score	128K	0.13	0.27
Scene Feature	Subscores	128K	0.11	0.20

nents, as well as incorporating open-world perception and retrieval-guided supplementary perception, consistently reduces the collision rate. Although the deviation from human driving trajectories increases slightly, this does not imply a decline in planning quality. We further analyze the effects of varying the numbers of knowledge entries and candidate trajectories. As shown in the middle and lower sections of the table, increasing these quantities beyond a certain threshold provides minimal additional benefit, indicating diminishing returns.

Ablation on the Value Model. As shown in Table 4, we evaluate several architectures for the Value Model. Using BEV images as input processed by CNNs offers the advantage of not requiring model-specific training for each baseline, but yields weaker predictive performance. Introducing weighted averaging across disaggregated subscores effectively reduces prediction errors. The entry labeled *10K* represents our initial dataset version, which included only trajectories generated by HENet++. In the final version, we augmented the dataset with numerous randomly generated low-quality trajectories as negative samples, significantly increasing both scale and diversity.

5. Conclusion

This paper presents KnowVal, a new autonomous driving system that achieves visual–language reasoning through the synergistic interaction between open-world perception and knowledge retrieval. Specifically, we construct a comprehensive knowledge graph for autonomous driving that integrates traffic laws, defensive driving principles, and ethical considerations, along with an efficient LLM-based retrieval mechanism tailored to driving contexts. To enable value-aligned decision-making, we curate a human-preference dataset and train a Value Model that provides an interpretable foundation for trajectory evaluation. Extensive experiments demonstrate that KnowVal substantially enhances planning performance while maintaining full compatibility with existing methods, achieving the lowest collision rate on nuScenes and establishing state-of-the-art performance on Bench2Drive.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 7

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017. 2

[3] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. 3

[4] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025. 2, 7

[5] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024. 3, 5

[6] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 2, 6

[7] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *CVPR*, 2023. 2, 6, 7

[8] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023. 2, 7

[9] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 2, 7

[10] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *NeurIPS*, 2024. 2, 7

[11] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *ICLR*, 2025. 2, 7

[12] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, 2023. 2, 6, 7

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020. 3

[14] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *ICLR*, 2025. 6

[15] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 2

[16] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, 2025. 2, 6, 7

[17] Zhiwei Lin and Yongtao Wang. Vl-sam-v2: Open-world object detection with general and specific query fusion. *arXiv preprint arXiv:2505.18986*, 2025. 4

[18] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *EMNLP*, 2023. 3

[19] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *CVPR*, 2025. 2, 7

[20] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*, 2023. 3

[21] Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in dit. *arXiv preprint arXiv:2504.15009*, 2025. 8

[22] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *CVPR*, 2025. 2, 6

[23] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *ICRA*, 2025. 2, 6

[24] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023. 6

[25] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024. 2, 6

[26] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *NeurIPS*, 2022. 2, 7

[27] Zhongyu Xia, Jishuo Li, Zhiwei Lin, Xinhao Wang, Yongtao Wang, and Ming-Hsuan Yang. Openad: Open-world autonomous driving benchmark for 3d object detection. *arXiv preprint arXiv:2411.17761*, 2024. 4

[28] Zhongyu Xia, Zhiwei Lin, Yongtao Wang, and Ming-Hsuan Yang. Henet++: Hybrid encoding and multi-task learning for 3d perception and end-to-end autonomous driving. *arXiv preprint arXiv:2511.07106*, 2025. 2, 6

[29] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *ICLR*, 2023. 3

[30] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023. 2, 7

[31] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *CVPR*, 2025. 2, 6

[32] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024. 2, 6

A. Qualitative Analysis of the Value Model

To better illustrate the role of our proposed and trained Value Model, Figure 7 showcases several examples and visualizes ego vehicle trajectories across two scenarios, where trajectories 2 and 4 were manually constructed as *negative samples* and the black trajectory in the center represents the ego vehicle. In the first case (top), the first trajectory successfully avoids pedestrians and the leading vehicle, whereas the trajectory in the second image heads directly toward them. A comparison of the predicted scores for knowledge entries 1, 3, and 5 demonstrates that the Value Model has successfully learned the concepts of ”maintaining a safe following distance” and ”respecting and yielding to pedestrians.”

In the second scenario (bottom, images 3 and 4), a large truck is present ahead of the ego vehicle. According to defensive driving principles, one should maintain a distance from large, risky vehicles. The comparison of predicted scores for knowledge entries 1 and 4 indicates that the Value Model has effectively learned the concepts of ”maintaining a safe following distance” and ”avoiding large vehicles.” In both scenarios, the scores predicted by our Value Model closely align with the GT.

The scores predicted by our value model in both scenarios align closely with the ground truth. This demonstrates the model’s ability to effectively gauge knowledge compliance across various contexts and assign reasonable scores accordingly.

B. Additional Implementation Details

B.1. Prompt used for scoring trajectories with MLLM

```
# ROLE
You are an expert autonomous driving
    auditor. Your analysis is meticulous,
    objective, and follows a strict logical
    process. Your primary goal is to
    identify and quantify risk.

# TASK
For each of the prioritized driving rules
    provided below, you MUST follow a
    mandatory **4-Step Critical Evaluation
    Process**. Your final score for each
    rule is a direct consequence of this
    process. Do not skip any steps.

# PRIORITIZED RULES TO EVALUATE
{formatted_rules}

# MANDATORY 4-STEP CRITICAL EVALUATION
    PROCESS
For each rule, you must internally answer
    these questions in order:

**Step 1: Applicability Check**
*   **Question:** Is this rule relevant to
    the current scene? Can it possibly be
    applied or violated based on the visual
    context?
*   **Outcome:**
    *   If **NO** (e.g., a traffic light
    rule when no traffic lights are visible)
    , the process stops here. Your
    conclusion is "Not Applicable". **The
    score MUST be 1.0.**
    *   If **YES**, proceed to Step 2.

**Step 2: Evidence Check**
*   **Question:** Does the image provide
    any visual information (positive or
    negative) to judge adherence to this
    applicable rule?
*   **Outcome:**
    *   If **NO** (e.g., a rule about
    checking blind spots, but the driver’s
    actions are not visible), the process
    stops here. Your conclusion is "
    Applicable, but no evidence". **The
    score MUST be 0.0.**
    *   If **YES**, proceed to Step 3.

**Step 3: Adherence Check**
*   **Question:** Based on the available
    evidence, is the ego vehicle fully and
    correctly adhering to the rule?
*   **Outcome:**
    *   If **YES**, the process stops here.
     Your conclusion is "Perfect Adherence".
```

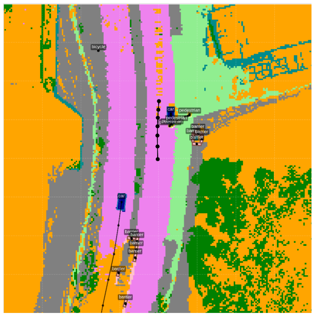
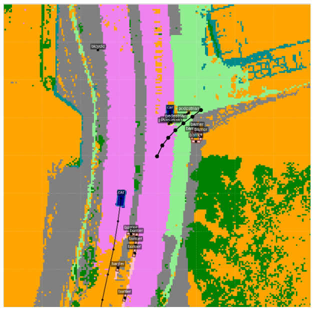
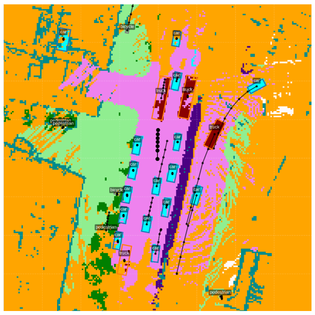
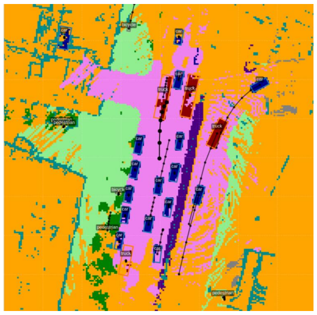
Perception Results and Input Trajectories to the Value Model	Retrieved Knowledge Entries	Prediction	GT
	1. Yield to pedestrians and non-motor vehicles.	0.7	1.0
	2. Do not honk to rush vulnerable pedestrians, including the elderly, disabled, or pregnant women.	0.4	0.0
	3. Maintain a safe following distance at all times to prevent rear-end collisions.	1.0	1.0
	4. Treat unexplained slow driving or frequent braking as a risk. Increase distance and overtake when safe.	0.9	0.9
	5. Watch for sudden crossings, especially on roads without crosswalks. Cover the brake upon seeing pedestrians and be ready to stop.	0.9	0.9
	1. Yield to pedestrians and non-motor vehicles.	-0.5	-0.7
	2. Do not honk to rush vulnerable pedestrians, including the elderly, disabled, or pregnant women.	0.3	0.0
	3. Maintain a safe following distance at all times to prevent rear-end collisions.	-0.7	-0.9
	4. Treat unexplained slow driving or frequent braking as a risk. Increase distance and overtake when safe.	-0.3	-0.6
	5. Watch for sudden crossings, especially on roads without crosswalks. Cover the brake upon seeing pedestrians and be ready to stop.	-0.5	-0.7
	1. Large vehicles are heavy, stop slowly, and have blind spots. Always maintain a safety gap far exceeding that of regular cars.	0.4	0.8
	2. Even with a green light, slow down and check the intersection. Be alert for sudden cross-traffic or pedestrians violating signals.	0.8	1.0
	3. In traffic congestion, do not block the intersection.	0.6	0.9
	4. Maintain a safe following distance sufficient for emergency braking.	0.8	1.0
	5. Slow down and proceed with caution when pedestrians or non-motorized vehicles are present on the roadside.	0.9	1.0
	1. Large vehicles are heavy, stop slowly, and have blind spots. Always maintain a safety gap far exceeding that of regular cars.	-0.7	-1.0
	2. Even with a green light, slow down and check the intersection. Be alert for sudden cross-traffic or pedestrians violating signals.	0.6	1.0
	3. In traffic congestion, do not block the intersection.	0.5	0.8
	4. Maintain a safe following distance sufficient for emergency braking.	-0.6	-0.9
	5. Slow down and proceed with caution when pedestrians or non-motorized vehicles are present on the roadside.	0.8	1.0

Figure 7. **Visualization of future state evaluation results using the proposed Value Model.** Displayed are BEV of the future state, retrieved knowledge entries, and Prediction vs. GT Scores. The comparison between compliant (Rows 1, 3) and risky (Rows 2, 4) trajectories demonstrates that our Value Model can evaluate future states against knowledge.

~~~
    **The score MUST be 1.0.**
    *   If **NO**, there is a violation.
    Proceed to Step 4 to determine its
    severity.

**Step 4: Risk Assessment (FOR VIOLATIONS
    ONLY)**
*   **Question:** How severe is the
    immediate risk created by this violation
    ?
*   **Outcome:** Assign a risk level, which
     then determines the score range.
    *   **Negligible Risk:** A technical
    error with no immediate safety impact (e
    .g., stopping 10cm past the stop line
    with no cross-traffic). **Score: -0.1 to
     -0.2**.
    *   **Low Risk:** A poor driving habit
    or minor violation that increases
    potential risk but doesn't create
    immediate danger (e.g., late signaling
    in light traffic). **Score: -0.3 to
    -0.4**.
    *   **Moderate Risk:** A clear unsafe
    action that significantly increases risk
     (e.g., tailgating, changing lanes
    without signal near other cars). **Score
    : -0.5 to -0.7**.
    *   **High Risk:** A critical error
    that creates a probable chance of
    collision or puts others in immediate
    danger (e.g., failing to yield to a
    pedestrian entering a crosswalk, running
     a stale yellow light). **Score: -0.8 to
     -1.0**.

# CRITICAL INSTRUCTIONS
- **AVOID LAZY SCORING:** Do not default to
     1.0 or 0.0 unless the 4-step process
    explicitly leads you there. The goal is
    to differentiate between different
    levels of non-compliance.
- **JUSTIFY YOUR SCORE:** The `conclusion`
    and the `risk_level` must logically and
    transparently lead to the final `score`.
- **BE SPECIFIC:** In your evidence
    description, refer to specific objects
    and actions in the scene.

# OUTPUT FORMAT
You MUST respond with a single JSON object.
     The object must have a single key "
    evaluations", which is a list. Each item
     in the list must adhere to the
    following structure:
1.  A "rule" key with the original rule
    text.
2.  A "reasoning" object containing four
    keys:
    - "positive_evidence": Visual cues that
     the rule is being followed. State "None
    " if none.
    - "negative_evidence": Visual cues that
     the rule is being violated. State "None
    " if none.
    - "risk_level": A string describing the
     risk for violations ("Negligible", "Low
    ", "Moderate", "High"). For non-
    violations, state "N/A".
    - "conclusion": A summary of your 4-
    step analysis, explicitly stating the
    final judgment (e.g., "Not Applicable",
    "Perfect Adherence", "Moderate Risk
    Violation").
3.  A "score" key, a float number that is a
     direct consequence of your `conclusion`
     and `risk_level`.

Example of your required output format:
```json
{{
  "scene_description": "A vehicle is
    approaching an intersection with a
    pedestrian waiting to cross at a marked
    crosswalk. The traffic light is green.",
  "evaluations": [
    {{
      "rule": "Yield to pedestrians at the
    crosswalk.",
      "reasoning": {{
        "positive_evidence": "The vehicle
    is slowing down as it approaches the
    crosswalk.",
        "negative_evidence": "The vehicle
    does not come to a complete stop and
    proceeds through the crosswalk, forcing
    the pedestrian to wait.",
        "risk_level": "High",
        "conclusion": "High Risk Violation.
     The rule is applicable and evidence is
    clear. The vehicle failed to yield to a
    pedestrian ready to cross, creating
    immediate danger."
      }},
      "score": -0.8
    }},
    {{
      "rule": "Obey traffic signals.",
      "reasoning": {{
        "positive_evidence": "The traffic
    light for the vehicle's direction of
    travel is green, and the vehicle is
    proceeding through the intersection.",
        "negative_evidence": "None",
        "risk_level": "N/A",
~~~

```
        "conclusion": "Perfect Adherence.
    The rule is applicable, and the vehicle
    is correctly following the green light
    signal."
      }},
      "score": 1.0
    }},
    {{
      "rule": "No Honking.",
      "reasoning": {{
        "positive_evidence": "None",
        "negative_evidence": "None",
        "risk_level": "N/A",
        "conclusion": "Applicable, but no
    evidence. The rule is relevant in a city
     scene, but there is no visual or
    contextual information to determine if
    the horn was used."
      }},
      "score": 0.0
    }},
    {{
      "rule": "Stop at Railroad Crossing.",
      "reasoning": {{
        "positive_evidence": "None",
        "negative_evidence": "None",
        "risk_level": "N/A",
        "conclusion": "Not Applicable.
    There are no railway tracks or crossings
     visible in the scene."
      }},
      "score": 1.0
    }}
  ]
}}"""
```

Listing 1. The prompt used for constructing GT of the Value Model Dataset

B.2. Prompt for keyword extraction in the retrieval stage

```
---Goal---
Given a text document that is potentially
    relevant to this activity and a list of
    entity types, identify all entities of
    those types from the text and all
    relationships among the identified
    entities.
Use {language} as output language.

---Steps---
1. Identify all entities. For each
    identified entity, extract the following
     information:
- entity_name: Name of the entity, use same
     language as input text. If English,
    capitalized the name.
- entity_type: One of the following types:
    [{entity_types}]
- entity_description: Comprehensive
    description of the entity's attributes
    and activities
Format each entity as ("entity"{
    tuple_delimiter}<entity_name>{
    tuple_delimiter}<entity_type>{
    tuple_delimiter}<entity_description>)

2. From the entities identified in step 1,
    identify all pairs of (source_entity,
    target_entity) that are *clearly related
    * to each other.
For each pair of related entities, extract
    the following information:
- source_entity: name of the source entity,
     as identified in step 1
- target_entity: name of the target entity,
     as identified in step 1
- relationship_description: explanation as
    to why you think the source entity and
    the target entity are related to each
    other
- relationship_strength: a numeric score
    indicating strength of the relationship
    between the source entity and target
    entity
- relationship_keywords: one or more high-
    level key words that summarize the
    overarching nature of the relationship,
    focusing on concepts or themes rather
    than specific details
Format each relationship as ("relationship
    "{tuple_delimiter}<source_entity>{
    tuple_delimiter}<target_entity>{
    tuple_delimiter}<
    relationship_description>{
    tuple_delimiter}<relationship_keywords>{
    tuple_delimiter}<relationship_strength>)

3. Identify high-level key words that
    summarize the main concepts, themes, or
    topics of the entire text. These should
    capture the overarching ideas present in
     the document.
Format the content-level key words as ("
    content_keywords"{tuple_delimiter}<
    high_level_keywords>)

4. Return output in {language} as a single
    list of all the entities and
    relationships identified in steps 1 and
    2. Use **{record_delimiter}** as the
    list delimiter.

5. When finished, output {
    completion_delimiter}
```

```
######################
---Examples---
######################
{examples}

#############################
---Real Data---
######################
Entity_types: [{entity_types}]
Text:
{input_text}
######################
Output:
```

Listing 2. The prompt used for retrieval

C. About the Knowledge Graph

A local view of the knowledge graph is shown in Figure 8. Currently, the knowledge graph contains 1,324 nodes and 2,785 edges.

D. Sample of the Value Model Dataset

Some samples from the value model dataset are shown in Figure 7. We will make the entire dataset and the model publicly available in the future.

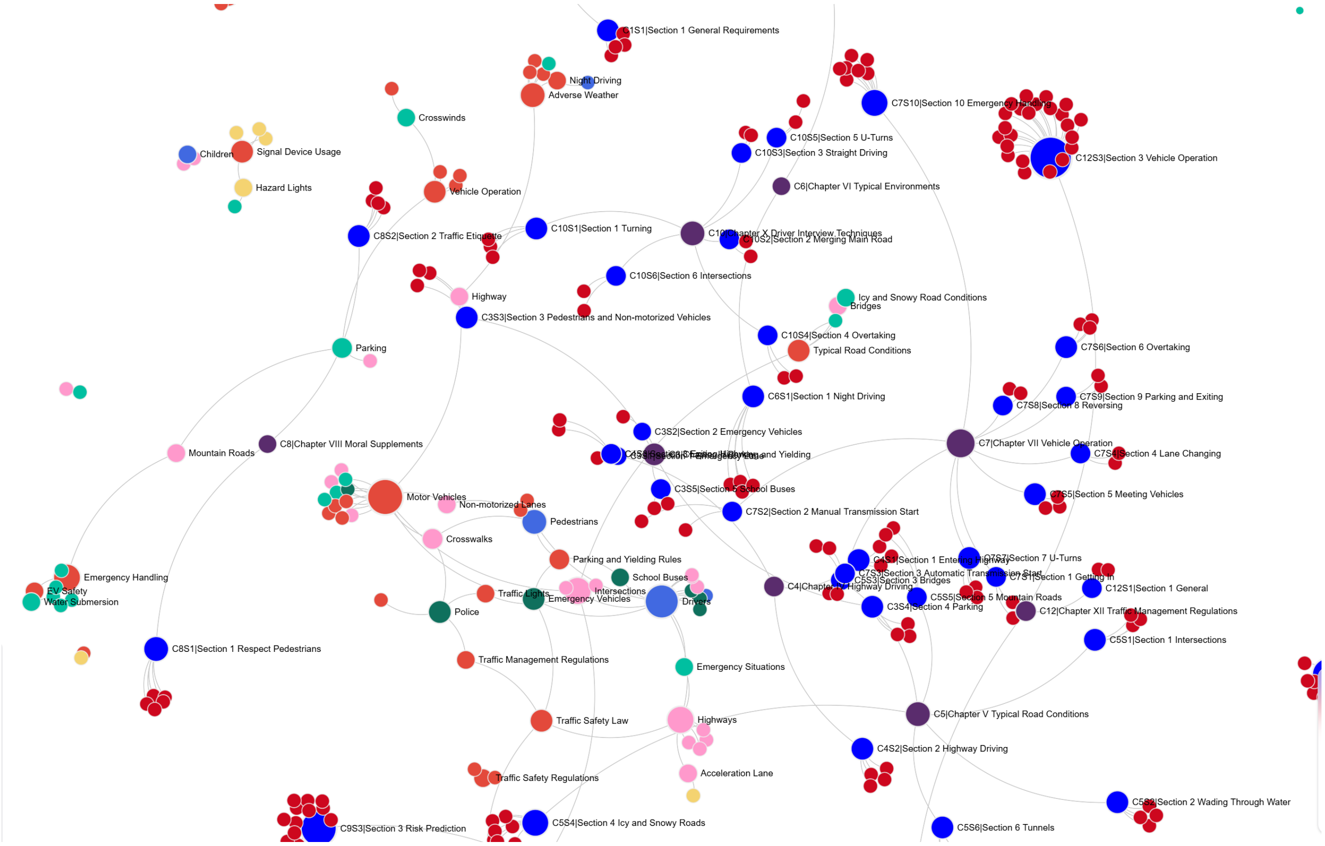

C1S1|Section 1 General Requirements
Night Driving
Adverse Weather
Crosswinds
Children
Signal Device Usage
Hazard Lights
Vehicle Operation
C8S2|Section 2 Traffic Etiquette
C10S1|Section 1 Turning
C10S6|Section 6 Intersections
Highway
C3S3|Section 3 Pedestrians and Non-motorized Vehicles
Parking
Mountain Roads
C8|Chapter VIII Moral Supplements
Motor Vehicles
Non-motorized Lanes
Pedestrians
Crosswalks
Parking and Yielding Rules
School Buses
Traffic Lights
Intersections
Emergency Vehicles
Drivers
Police
Emergency Handling
EV Safety
Water Submersion
C8S1|Section 1 Respect Pedestrians
Traffic Management Regulations
Emergency Situations
Traffic Safety Law
Highways
Traffic Safety Regulations
Acceleration Lane
C9S3|Section 3 Risk Prediction
C5S4|Section 4 Icy and Snowy Roads
C7S10|Section 10 Emergency Handling
C12S3|Section 3 Vehicle Operation
C10S5|Section 5 U-Turns
C10S3|Section 3 Straight Driving
C6|Chapter VI Typical Environments
C10|Chapter X Driver Interview Techniques
C10S2|Section 2 Merging Main Road
Icy and Snowy Road Conditions
Bridges
C10S4|Section 4 Overtaking
Typical Road Conditions
C7S6|Section 6 Overtaking
C6S1|Section 1 Night Driving
C7S9|Section 9 Parking and Exiting
C7S8|Section 8 Reversing
C3S2|Section 2 Emergency Vehicles
C7|Chapter VII Vehicle Operation
C7S4|Section 4 Lane Changing
C3S5|Section 5 School Buses
C7S2|Section 2 Manual Transmission Start
C7S5|Section 5 Meeting Vehicles
C4S1|Section 1 Entering Highway
C7S7|Section 7 U-Turns
C7S3|Section 3 Automatic Transmission Start
C7S1|Section 1 Getting In
C5S3|Section 3 Bridges
C4|Chapter IV Highway Driving
C12S1|Section 1 General
C5S5|Section 5 Mountain Roads
C3S4|Section 4 Parking
C12|Chapter XII Traffic Management Regulations
C5S1|Section 1 Intersections
C5|Chapter V Typical Road Conditions
C4S2|Section 2 Highway Driving
C5S2|Section 2 Wading Through Water
C5S6|Section 6 Tunnels

Figure 8. **Local view of the knowledge graph.**