2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Beyond Scalar Rewards: Distributional Reinforcement Learning with Preordered Objectives for Safe and Reliable Autonomous Driving

Ahmed Abouelazm*[1,2], Jonas Michel*[2], Daniel Bogdoll[1,2], Philip Schörner[1,2], and J. Marius Zöllner[1,2]

Abstract— **Autonomous driving involves multiple, often conflicting objectives such as safety, efficiency, and comfort. In reinforcement learning (RL), these objectives are typically combined through weighted summation, which collapses their relative priorities and often yields policies that violate safety-critical constraints. To overcome this limitation, we introduce the Preordered Multi-Objective MDP (Pr-MOMDP), which augments standard MOMDPs with a preorder over reward components. This structure enables reasoning about actions with respect to a hierarchy of objectives rather than a scalar signal. To make this structure actionable, we extend distributional RL with a novel pairwise comparison metric, Quantile Dominance (QD), that evaluates action return distributions without reducing them into a single statistic. Building on QD, we propose an algorithm for extracting optimal subsets, the subset of actions that remain non-dominated under each objective, which allows precedence information to shape both decision-making and training targets. Our framework is instantiated with Implicit Quantile Networks (IQN), establishing a concrete implementation while preserving compatibility with a broad class of distributional RL methods. Experiments in Carla show improved success rates, fewer collisions and off-road events, and deliver statistically more robust policies than IQN and ensemble-IQN baselines. By ensuring policies respect rewards preorder, our work advances safer, more reliable autonomous driving systems.**

I. Introduction

End-to-End (E2E) learning has emerged as a compelling paradigm for Autonomous Driving (AD), directly mapping raw sensory input to vehicle actions within a unified model [1]. By bypassing handcrafted intermediate stages, E2E approaches mitigate error accumulation and enable scalable, data-driven driving policies [2]. Unlike modular pipelines, which rely on carefully engineered perception and decision-making components, E2E systems reduce error propagation between modules and allow joint optimization of perception and control. This streamlines learning and lowers reliance on manual design [3].

While imitation learning (IL) is effective for acquiring basic driving skills, Reinforcement Learning (RL) offers distinct advantages by optimizing behavior through direct interaction with the environment [4]. By maximizing cumulative rewards, RL agents can adapt to the dynamic and uncertain conditions of real-world traffic [5]. At the core of this process lies the reward function, which specifies driving objectives such as safety, efficiency, and comfort, and thus directly governs the quality of the learned policy [6].

* These authors contributed equally to this work
[1]Authors are with the FZI Research Center for Information Technology, Germany name@fzi.de
[2]Authors are with the Karlsruhe Institute of Technology, Germany

Research Gap. Designing reward functions for AD is inherently complex, as they must capture multiple, often conflicting objectives, such as safety, efficiency, and comfort, while also reflecting their relative priorities [7]. A common practice is to collapse these objectives into a single scalar reward, typically through naive or weighted summation [5]. However, such formulations are difficult to tune, highly context-dependent, and have been shown to produce undesirable behaviors when trade-offs are misaligned, such as prioritizing comfort or efficiency at the expense of safety [8].

To address these deficiencies, research in Multi-Objective Reinforcement Learning (MORL) has proposed representing a reward as a vector of distinct objectives [9]. This allows the agent to learn separate value estimates per objective. However, decision-making still relies on weighted aggregation of these estimates [10], which limits the learned policy's ability to preserve the intended priority of objectives.

Alternative approaches introduce hierarchical rewards inspired by rulebooks [11], where relations among objectives are explicitly encoded [12], [13]. While these approaches provide interpretability and structured priorities at the reward-design level, agents are still trained with scalar rewards, restricting agents' ability to disentangle the contribution of each objective.

Therefore, a gap remains in developing RL agents that can semantically represent multiple objectives and enforce their relative priorities within the learning process itself. Such a formulation enables safety-critical objectives to guide both learning and decision-making more reliably, leading to safer driving behavior.

Contribution. This paper presents a framework that incorporates preorder relations, capturing the relative priority between objectives, directly into RL agents. In this way, objective priorities are respected during both training and inference. The key contributions of this work are:

- **Preordered Multi-Objective MDP (Pr-MOMDP):** We extend MOMDPs with preorder relations, providing a formulation that enables reasoning about actions with respect to prioritized objectives.
- **Quantile-based action comparison:** Building on the Pr-MOMDP formulation, we propose Quantile Dominance (QD), a distributional metric that compares full return distributions to derive pairwise action relations.
- **Optimal subsets for decision-making:** Leveraging QD, we extract optimal action subsets, non-dominated actions under each objective, and integrate them into both action selection and training updates, ensuring higher-priority objectives consistently shape policy learning.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

II. RELATED WORK

The integration of multi-objective and hierarchical reward structures into RL policies is a critical area of research. While significant progress has been made, major challenges remain. Experiments in the complex domain of AD have revealed that current approaches insufficiently respect reward structures, leading to undesirable behavior [8]. In this section, we first introduce classical reward structures and their challenges in the context of AD, and subsequently introduce prior work in MORL and Hierarchical Reinforcement Learning (HRL) that aims to handle such complex reward structures.

A. Reward Design in Autonomous Driving

AD is a complex domain with a multitude of often conflicting objectives [7]. This makes it challenging to manually design reward functions and can often lead to insufficient performance [14]. Knox et al. [8] identified several challenges in the manual design of reward functions, such as undesired risk tolerance or preference orderings that do not align with human judgment. These findings are confirmed by a large-scale survey by Abouelazm et al. [7], highlighting that most reward functions in the literature utilize different individual reward terms but aggregate them into a scalar output, eliminating context awareness and relative ordering.

To avoid error-prone manual reward designs, another line of work proposes the automated generation of a reward function based on Large Language Models (LLMs) [15]. However, this approach still collapses multiple reward terms into a single scalar, not solving the core issue of classical reward structures. Finally, some works address the runtime adoption of driving behaviors by introducing priors on driving aspects, such as comfort or aggressiveness [16], [17]. Here, classically engineered reward terms are combined with prior conditions so agents can display different behaviors during inference without re-training. These approaches emphasize certain aspects of total reward rather than integrating hierarchies to address complex reward structures.

B. Multi-Objective RL and Hierarchical Rewards.

As classical reward structures struggle to address scenarios that require hierarchical or multi-objective reward signals, this section presents advancements in MORL and HRL.

Rather than just decomposing components of the reward function, several works adapt model architectures by introducing multi-branch networks for individual reward components [10], [18]–[22]. While these concepts have demonstrated performance improvements and can be used to dynamically adjust weights during runtime [23], [24], an artificial bottleneck is introduced by merging the resulting Q-values into a singular value for training.

Differently, Deshpande et al. [9] use a Deep Q-Network (DQN) per reward objective and generate a list of acceptable actions for each. However, the action selection is based on sequential filtering and ordering, which is error-prone and cannot capture the full range of relations between objectives. Several works have combined distributional RL with multi-dimensional rewards [25]–[28]. However, they utilize simple, unstructured rewards, which are not suitable to address complex real-world applications such as AD.

Only a few works integrate complex reward structures in RL applications. Bogdoll et al. [12] proposed a rulebook-based [11] and situation-aware reward function that showed performance improvements in traffic scenarios that required controlled rule exceptions. Abouelazm et al. [13] similarly designed rulebook-based rewards with a novel risk term and a normalization scheme that assigns weights according to the hierarchy level of each reward term. However, both approaches collapse the structured reward into a single scalar, limiting their ability to fully exploit the hierarchy.

Unlike prior approaches that collapse objectives into a single scalar (Fig. 2a), aggregate value estimates without preserving preorder (Fig. 2b), or rely on unstructured rewards, our approach preserves preorder between objectives, leverages distributional value estimates for robust action comparison, and encodes reward hierarchies directly into the learning process.

III. METHODOLOGY

In this section, we formalize our approach, illustrated in Fig. 2c, to incorporating reward preorder into RL. We first extend the MOMDP with a preorder relation, referred to as precedence among objectives, to capture hierarchical structure (Section III-A). We then introduce a distributional metric for action comparison and a preorder-guided action selection framework that adapts both architecture and training to respect priorities (Section III-C).

A. Problem Formulation

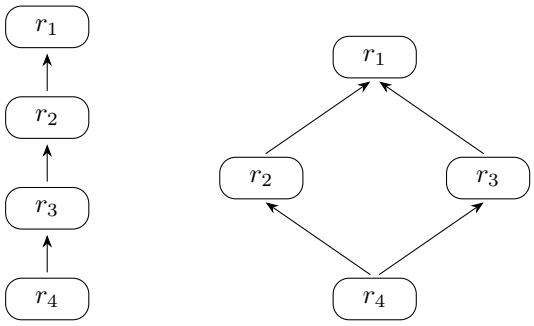
MORL is typically formalized through Multi-objective markov decision process (MOMDP), $\mathcal{M}_{\text{MOMDP}} = \langle S, A, P, \mathcal{R}, \gamma \rangle$, where S is a finite set of states, A a finite set of actions, and $P(s' \mid s, a)$ denotes the transition probability from state s to state s' under action a.

For N objectives, the reward function is a vector given by $\mathcal{R} : S \times A \to \mathbb{R}^N$. For any $(s, a) \in S \times A$, the vectorized reward is realized as $\mathcal{R}(s, a) = \{r_i(s, a)\}_{i=1}^{N}$, where $r_i(s, a)$ denotes the reward for objective i. The discount factors are similarly defined as $\gamma = \{\gamma_i\}_{i=1}^{N}$.

Existing MOMDP formulations typically handle multiple objectives either by treating all reward components as equally weighted and aggregating them into a single signal [29] or by enforcing a strict lexicographic order [9]. Both approaches impose rigid constraints that limit the framework's ability to capture more flexible relations among objectives.

To address this limitation, we introduce the Preordered MOMDP (Pr-MOMDP), an extension of MOMDP that incorporates a pre-order relation $\succeq$ over reward components. This extension preserves the vectorized reward structure while enabling comparisons that respect the hierarchy among objectives. In contrast to rulebooks [11], which use $\preceq$ because they operate on costs to be minimized, we employ $\succeq$ since our formulation is reward-based and maximizes returns. The proposed formulation of Pr-MOMDP is given in Eq. 1.

$$\mathcal{M}_{\text{Pr-MOMDP}} = \mathcal{M}_{\text{MOMDP}} + \langle \succeq \rangle = \langle S, A, P, \mathcal{R}, \gamma, \succeq \rangle \quad (1)$$



(a) Lexicographic Reward (b) Partially Ordered Reward

Fig. 1: Examples of lexicographic and partial order rewards

For any $r_i, r_j \in \mathcal{R}$, the relation $r_i \succeq r_j$ indicates that the reward component r_i has a higher priority than r_j. The introduction of a pre-order allows reward relations to be represented flexibly as directed graphs. Figure 1 illustrates two instances: a total order (lexicographic) with $r_1 \succeq r_2 \succeq r_3 \succeq r_4$, and a partial order in which r_2 and r_3 remain incomparable. Such flexibility is essential for capturing both strict hierarchies and more general priority structures that arise in multi-objective decision-making.

To address the complexities of AD, we extend the formulation in Eq. 1 from the fully observable case to the more challenging partially observable setting. Here, the agent interacts with the environment through observations $o \in O$ generated by a sensor model. This extension leaves the preorder relation $\succeq$ unaffected, as prioritization among reward components is independent of the observation process.

B. Action Relations from Rewards Preorder

The introduction of a precedence relation not only structures the reward components themselves but also induces a relational semantics among actions. Building on the relations introduced in [30], we adapt them to the proposed Pr-MOMDP setting: given two actions $a, a' \in A$ with corresponding reward components $\mathcal{R}(s,a), \mathcal{R}(s,a') \in \mathbb{R}^N$ and a preorder relation $\succeq$ over objectives, we define:

- **Dominance:** a dominates a' if there exists an objective r_j satisfying $r_j(s,a) > r_j(s,a')$, and for any objective with $r_i(s,a') > r_i(s,a)$, it holds that $r_j \succ r_i$ under $\succeq$.
- **Indifference:** a and a' are indifferent if neither a dominates a' nor a' dominates a.
- **Incomparability:** a and a' are incomparable if there exist objectives r_i and r_j such that $r_i(s,a) > r_i(s,a')$ and $r_j(s,a') > r_j(s,a)$, and neither objective is comparable (ordered above the other) under $\succeq$.

Precedence-based relations provide a meaningful way to compare actions in terms of their reward vectors, but RL agents do not act directly on rewards. Instead, decisions are guided by value functions that estimate expected return over time. To enable agents to benefit from the semantic structure of rewards, we address how precedence-based action relations can be extended into the value-function space. The next section develops a comparison algorithm that integrates these relations into learning, allowing objective hierarchies to guide action evaluation and policy learning.

C. Preorder-guided Action Selection

1) Agent Architecture: Representing multiple value functions within the agent architecture raises design challenges. Factored-state approaches [31] assign each objective to a separate subset of the state, but this requires handcrafted features and is infeasible when learning directly from raw sensor data. Using the full state with separate networks per objective [9] avoids hand-design but duplicates computation and prevents objectives from benefiting from shared representations [32].

To address these limitations, we adopt a multi-head architecture: observations are encoded into a common latent representation that captures task-relevant features, which is then passed to multiple heads, as demonstrated in Fig. 2c. Each head outputs a value estimate for its corresponding objective $r_i \in \mathcal{R}$, allowing new objectives to be added simply by introducing an additional head. This design improves scalability, and leverages shared features while maintaining objective-specific value predictions.

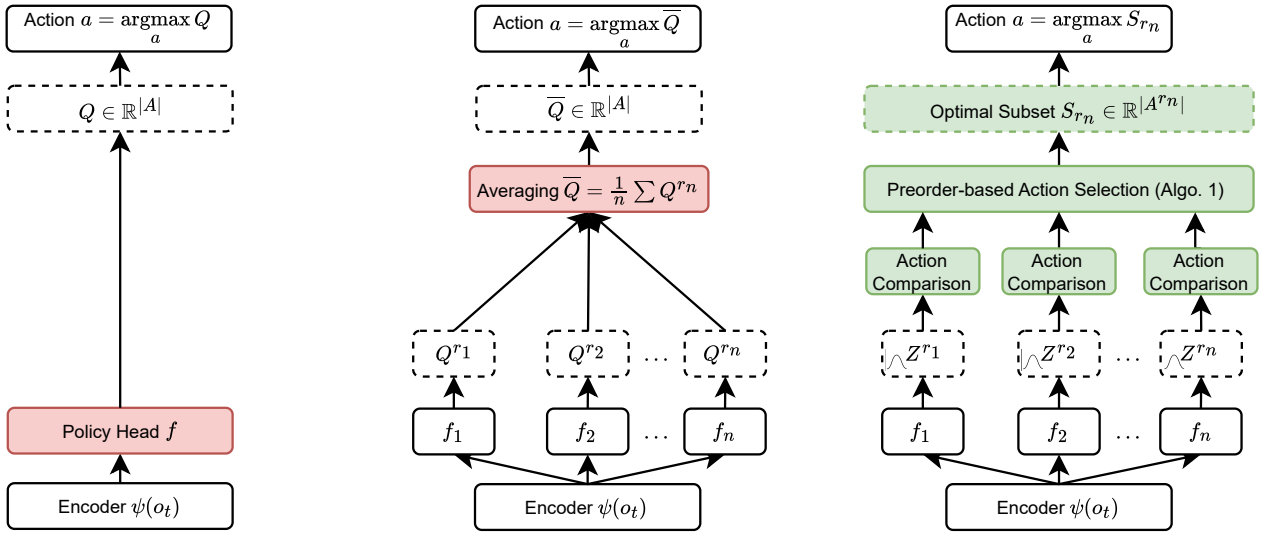
2) Value Functions Estimation: Strictly applying precedence at the level of value estimates raises important challenges. Previous rulebook approaches [33] rely on discrete, boolean comparisons that assume rewards can be evaluated as satisfied or violated. Value functions, by contrast, often have large magnitudes, are noisy, and fluctuate during training. Enforcing strict prioritization in this setting can lead to undesirable outcomes. For example, strict prioritization may lead the agent to favor a negligible gain in clearance over significant progress, resulting in overly conservative behavior such as remaining stationary. Such brittleness highlights the need for a more tolerant evaluation mechanism that accounts for the uncertainty in value estimates and can capture precedence in a distributional form.

To address these limitations, we adopt a distributional RL approach. Specifically, we use quantile-based value function estimates inspired by Implicit Quantile Network (IQN) [34], combined with a distribution-aware metric for pairwise action comparison (Section III-C.3) to enable more tolerant evaluation of actions under the same objective r_i. These comparisons then inform a preorder-based action selection algorithm (Algo. 1) that maintains, for each objective, the subset of non-dominated actions consistent with the hierarchy, denoted the optimal subset.

IQN models the entire quantile function, treating the quantiles $\tau \in [0,1]$ as a continuous random variable. This allows the network to approximate the inverse cumulative distribution function (inverse CDF F^{-1}) of the return distribution, as expressed in Eq. 2. For each objective r_i, the network outputs a matrix $Z^{r_i}(o_t, a) \in \mathbb{R}^{|\tau| \times |A|}$, where each row corresponds to a sampled quantile τ and each column to an action a.

$$Z^{r_i}_{\tau}(o_t, a) \;\approx\; F^{-1}_{Z^{r_i}(o_t,a)}(\tau), \quad \tau \sim \mathcal{U}[0,1] \tag{2}$$

3) Distribution-aware Pairwise Comparison: This section focuses on comparing actions using the full return distribution of a reward component r_i. Previous works collapse the distribution into a single statistic, such as conditional value-at-risk (CVaR) [34] or mean variance (MV) [35], thereby



(a) Single-Head architectures [12], [13], where all objectives are entangled in a single policy head f without the ability to separate them.

(b) Multi-Head architectures [10], [18], which learns one head per objective r_i but collapses decision-making to the mean value estimate.

(c) Our Pr-IQN with a novel action comparison and selection algorithm to utilize all available information and preserve a given preorder.

Fig. 2: Comparison of two classical architectures (a, b) with our Pr-IQN approach (c), shown during inference given observations o_t. Information bottlenecks are highlighted in red and novel components for full information utilization in green. Compared to classical approaches, Pr-IQN leverages distributions Z^{r_n} to select actions that respect a given preorder.

discarding distributional structure and increasing sensitivity to noise. In contrast, we propose quantile dominance (QD), a distribution-aware metric that compares action distributions to a quantile-wise ideal reference.

To ensure such comparisons are robust to estimation noise and consistent across objectives, we enforce scale invariance by normalizing quantile estimates per objective using Z-score normalization, denoted by $\tilde{Z}^{r_i}$. We then define the ideal distribution as the maximum return across all actions at each quantile τ_k, as shown in Eq. 3.

$$Z^{*,r_i}_{\tau_k} = \max_{a \in A} \tilde{Z}^{r_i}_{\tau_k}(a) \tag{3}$$

Accordingly, the quality of an action is measured by its Wasserstein-1 distance to this ideal profile, as given in Eq.4. Since smaller distances indicate stronger alignment with the quantile-wise optimum, we define the scalar action score $\text{score}^{r_i}(a) = -\widehat{W}_1^{r_i}(a)$, such that higher values correspond to stronger quantile dominance.

$$\widehat{W}_1^{r_i}(a) = \frac{1}{|\tau|} \sum_{\tau_k \in \tau} \left| \tilde{Z}^{r_i}_{\tau_k}(a) - Z^{*,r_i}_{\tau_k} \right|. \tag{4}$$

Finally, we define QD between two actions as demonstrated in Eq. 5, which quantifies the directional difference in quantile dominance. By construction, QD is asymmetric, i.e., $\text{QD}_{a \to a'} \neq \text{QD}_{a' \to a}$.

$$\text{QD}^{r_i}_{a \to a'} = \text{score}^{r_i}(a) - \text{score}^{r_i}(a'). \tag{5}$$

To avoid overly strict action comparisons, we introduce a tolerance parameter $\epsilon_{r_i} \in \mathbb{R}$ for each objective. Two actions a and a' are deemed *indifferent* if $|\text{QD}^{r_i}_{a \to a'}| \leq \epsilon_{r_i}$. Action a *dominates* a' if $\text{QD}^{r_i}_{a \to a'} > \epsilon_{r_i}$, and is *dominated by* a' otherwise. The QD procedure provides a principled way to assign pairwise relations between actions under a single objective r_i. In the next section, we extend it from individual objectives to the full pre-order structure over rewards.

4) Preorder Traversal and Action Selection: In contrast to rulebook planners [30] that require exhaustive evaluation over all objectives and actions, our algorithm is more efficient. It operates on the fixed action space of the RL agent, runs in linear time with respect to the number of reward components N, and yields optimal subsets at each level of the hierarchy, enabling direct use in agent training.

Algorithm 1 evaluates action relations while preserving the reward preorder. We apply a topological ordering based on the reward precedence [30], so that the parents of each reward r_i (i.e., directly connected higher-priority rewards) are always evaluated before r_i. At each step, dominance relations established at parent objectives are first inherited: if an action pair (a, a') is already determined to be dominated, dominating, or incomparable, this relation cannot be overridden by a lower-priority objective. In addition, we inherit an action optimal subset $\mathcal{S}^{\uparrow}$ via $\text{Agg}(\cdot)$, which aggregates parent survivor sets, and removes only actions effectively dominated by a surviving non-conflicting dominator.

Only indifferent (undecided) pairs are passed forward for evaluation, where they are compared using the QD operator (III-C.3), to yield local dominance relations. The inherited and local outcomes are merged to update the global dominance structure, while conflicts are filtered out. Finally, the optimal action subset $\mathcal{S}_{r_i} \subseteq \mathcal{S}^{\uparrow}$ is constructed, containing actions that are not strictly dominated under r_i. This stepwise filtering propagates precedence consistently through the preorder while pruning actions that fail higher-priority objectives. A key property of the algorithm is that each optimal subset of actions is guaranteed to be non-empty, ensuring that at least one feasible action remains available at every level of the hierarchy.

Algorithm 1: PREORDER ACTION SELECTION

Input: action set A; objectives $\mathcal{R}$ with preorder $\succeq$; parent map $\mathrm{Pa}(\cdot)$; quantile estimates $\{Z^{r_i}\}_{i=1}^{N}$; comparator $\mathrm{QD}(\cdot)$
Output: for each $r_i \in \mathcal{R}$: optimal subset $\mathcal{S}_{r_i}$

$\mathcal{L} \leftarrow \mathrm{TOPOLOGICALSORT}(\mathcal{R}, \succeq)$
for $r_i \in \mathcal{L}$ **do**
  (1) Inherit parent relations
  if $\mathrm{Pa}(r_i) = \varnothing$ **then**
    $\mathcal{S}^{\uparrow} \leftarrow A$; $Dom^{\uparrow} \leftarrow 0$; $DomBy^{\uparrow} \leftarrow 0$
  else
    $\mathcal{S}^{\uparrow} \leftarrow \mathrm{Agg}(\{\mathcal{S}_p\}_{p \in \mathrm{Pa}(r_i)})$
    $Dom^{\uparrow} \leftarrow \bigvee_{p \in \mathrm{Pa}(r_i)} Dom[p]$
    $DomBy^{\uparrow} \leftarrow \bigvee_{p \in \mathrm{Pa}(r_i)} DomBy[p]$
  (2) Construct update mask (only update indifferent pairs)
  $Mask \leftarrow \neg(Dom^{\uparrow} \vee DomBy^{\uparrow})$
  (3) Compare action pairs using QD
  $(Dom^{\mathrm{QD}}, DomBy^{\mathrm{QD}}) \leftarrow \mathrm{QD}(Z^{r_i}, Mask)$
  (4) Merge inherited and local
  $Dom[r_i] \leftarrow (\neg Mask \wedge Dom^{\uparrow}) \vee (Mask \wedge Dom^{\mathrm{QD}})$
  $DomBy[r_i] \leftarrow (\neg Mask \wedge DomBy^{\uparrow}) \vee (Mask \wedge DomBy^{\mathrm{QD}})$
  (5) Compute optimal subset at reward r_i
  $C \leftarrow Dom[r_i] \wedge DomBy[r_i]$
  $DomBy^{\downarrow} \leftarrow DomBy[r_i] \wedge \neg C$
  $\mathcal{S}_{r_i} \leftarrow \{\, a \in \mathcal{S}^{\uparrow} \mid \nexists\, a' \in \mathcal{S}^{\uparrow} :\ DomBy^{\downarrow}[a, a'] = 1 \,\}$
return $\{\mathcal{S}_{r_i}\}_{r_i \in \mathcal{R}}$

Legend: $\mathrm{Agg}(\cdot)$: aggregation of parent survivor sets; $Dom[a, a'] = 1$ if a dominates a'; $DomBy[a, a'] = 1$ if a is dominated by a'; $\bigvee$ = element-wise logical OR (over parent relations); $Mask$ = undecided action pairs mask; C = Incomparable action pairs
superscripts: $\uparrow$ = inherited from parents, QD = computed at r_i via quantile dominance, $\downarrow$ = dominated-by after incomparable removal

5) Preorder Informed Training and Inference: Preorder relations between reward components induce optimal action subsets at the value-function level. To leverage these sets during learning, we extend IQN [34] and denote the resulting algorithm as *Pr-IQN*. Conventional MORL approaches [9], [31] perform argmax-based target selection over the full action set for each objective, ignoring precedence relations among rewards. In contrast, Pr-IQN modifies the temporal difference (TD) [36] training targets to respect reward precedence. Specifically, we restrict the target selection for each objective r_i to the optimal action subset $\mathcal{S}_{r_i}$ obtained from Alg. 1, as defined in Eq. 6. This masking prevents selecting actions that achieve high return for r_i while violating higher-priority objectives. Accordingly, the TD error between quantiles (τ, τ'), denoted $\delta_t^{r_i,(\tau,\tau')}$, is computed as in Eq. 7, ensuring that value updates promote actions that optimize the current objective while respecting precedence constraints.

$$a_{t+1}^{*,r_i} = \arg\max_{a \in \mathcal{S}_{r_i}} \frac{1}{|\tau|} \sum_{\tau_k \in \tau} Z_{\tau_k}^{r_i}(o_{t+1}, a) \tag{6}$$

$$\delta_t^{r_i,(\tau,\tau')} = r_t^i + \gamma\, Z_{\tau'}^{r_i}(o_{t+1}, a_{t+1}^{*,r_i}) - Z_{\tau}^{r_i}(o_t, a_t) \tag{7}$$

During inference, the agent samples an action uniformly from the optimal subset associated with a leaf objective, following the approach in [30]. When the hierarchy contains multiple leaves, we introduce a virtual global leaf that aggregates their optimal subsets and guides action selection.

IV. EXPERIMENTAL SETUP

This section details the experimental setup, including the RL agent design, hierarchical reward structure, and urban traffic scenarios in CARLA [37]. We also outline baselines, ablations, and evaluation metrics to enable a systematic and fair comparison of performance.

A. RL Agent Description

We design a multimodal observation space that combines a front-facing RGB camera with resolution 128×128 and a LiDAR point cloud projected onto a 128×128 grid map with two vertical bins. Additionally, the agent is conditioned on high-level navigational commands [38] and on vehicle kinematics, including longitudinal and lateral velocities and accelerations. To encode this observation, we employ TransFuser [38], a transformer-based backbone that fuses image and LiDAR features into a shared latent representation.

For decision-making, we couple the RL agent with a Frenet-based planner [12], which generates trajectories consistent with road geometry. The agent outputs two discrete boundary conditions (v_f, d_f): v_f denotes the target velocity at the end of the planning horizon, and d_f the lateral displacement from the lane centerline. These conditions are used by the Frenet planner to construct a feasible trajectory.

B. Reward Hierarchy

The reward hierarchy illustrated in Fig. 3 organizes driving objectives according to their criticality for safe and reliable AD. *Safety* has the highest priority, as collision and off-road events are enforced as first-order constraints due to their catastrophic consequences [11]. The second level addresses *risk mitigation*, encouraging conservative driving behavior by maintaining clearance and proactively reducing collision likelihood [13]. Placing risk directly below safety ensures that near-miss situations are penalized before progress incentives can dominate. Below risk, *lane keeping* enforces compliance with road geometry, supporting both safety and predictability in mixed-traffic [33]. *Progress* follows, rewarding efficient route advancement and adherence to target velocity profiles. Finally, *comfort* is assigned the lowest priority as it primarily affects ride quality rather than immediate safety.

C. Traffic Scenarios

In this work, we focus on urban driving tasks where an autonomous agent must approach and cross unsignalized intersections. Such intersections are among the most safety-critical elements of road networks due to the absence of explicit right-of-way indicators and the need for implicit negotiation with other vehicles [39]. While our framework applies to a broad range of road scenarios, intersections provide a particularly demanding setting for evaluating risk-sensitive RL strategies.

Traffic scenarios are generated in CARLA [37], where training involves randomized configurations of static obstacles and traffic vehicles across multiple T-junctions and four-way intersections. Vehicle attributes such as geometry, speed, and lateral positioning are randomized to promote robustness

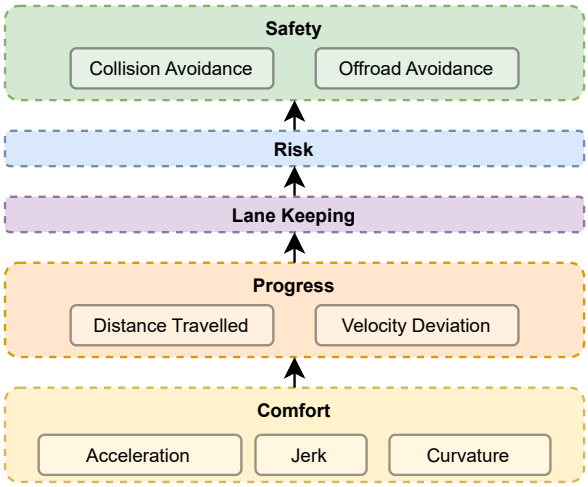


Fig. 3: The reward hierarchy with Safety as the highest priority, followed by Risk, Lane Keeping, Progress, and Comfort. This ordering guides the agent's decision-making to emphasize safety while balancing other objectives.

and generalization. For evaluation, we adopt a hold-out set consisting of one unseen T-junction and two unseen four-way intersections, ensuring that performance is assessed on layouts not encountered during training.

D. Baselines and Evaluation Metrics

We benchmark our RL framework against IQN [34], a widely used distributional RL baseline. To control for model capacity, we introduce an ensemble IQN variant with the same number of policy heads as our reward hierarchy, allowing us to disentangle gains from increased capacity and those from explicitly encoding semantic structure. Both baselines are trained using a weighted sum of the hierarchical reward components described in Section IV-B, following the weighting scheme of [13].

Additionally, we adopt the multi-objective approach of [10], [18], which learns one value head per objective r_i and selects actions using the mean value across objectives, denoted as mean aggregated IQN (MA-IQN). To isolate the contribution of each component of our framework, we conduct ablation studies examining the effect of allocating separate policy heads per objective, integrating hierarchical comparisons during training, and varying both the comparison method and threshold. We also evaluate the method under partial ordering, using a hierarchy in which risk and lane keeping are children of safety and precede progress.

To ensure a fair comparison, all agents are trained for the same number of steps using identical architectures and hyperparameters. Training is repeated with three random seeds, and each policy is evaluated over three runs to account for stochasticity in CARLA, following the protocol of [40]. Evaluation uses a hold-out set of intersection scenarios with varying traffic densities, defined as the ratio of active actors to the maximum allowed in the environment.

We evaluate performance using driving metrics and statistical reliability measures. Driving performance is measured using success rate (SR), off-road rate (OR), collision rate

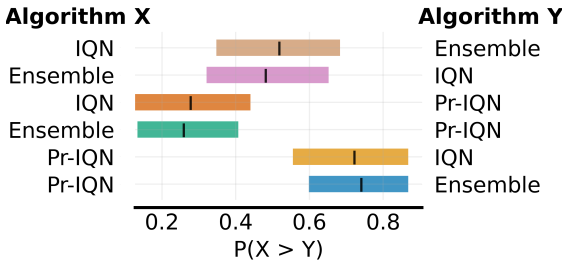


Fig. 4: Probability of improvement [41], quantifying the likelihood that an algorithm X (the left column) outperforms algorithm Y (the right column).

(CR), and route progress (RP), reported as mean $\pm$ standard deviation across all seeds and runs. We also assess the agent's ability to optimize individual reward components, reflecting alignment with the designed objectives. To complement these metrics, we use the RLiable library [41] to compute statistics such as the interquartile mean (IQM) and optimality gap.

V. Evaluation

Table II reports an ablation study analyzing the impact of the comparison metrics, the tolerance ϵ, and the integration of the preorder during training. MA-IQN performs noticeably worse since averaging value estimates across heads collapses the hierarchical ordering, allowing lower-priority improvements to compensate for violations of higher-priority objectives. Similarly, collapsing quantile estimates into scalar metrics such as CVaR or MV in Pr-IQN removes the distributional structure, leading to unreliable optimal action subsets.

Pr-IQN with QD consistently outperforms IQN and Ensemble across all traffic densities. Tightening the tolerance from $\epsilon = 0.4$ to $\epsilon = 0.2$ yields further gains by enforcing stricter adherence to the preorder. Incorporating the preorder during training further improves the success rate by $+3.3\%$ and $+2.5\%$ at densities 0.75 and 1.0. A partial preorder performs comparably to the total-order variant when the hierarchy is enforced during training, but is more sensitive to degradation when it is not, highlighting the importance of aligning the training objective with the decision structure.

Overall, our best configuration, Pr-IQN* (QD with total preorder enforced during training and $\epsilon = 0.2$), improves success rate by $(+7.7\%, +16.6\%, +20.3\%)$ over IQN and $(+11.0\%, +14.7\%, +13.9\%)$ over Ensemble-IQN at traffic densities 0.5, 0.75, and 1.0. These results show that preorder-guided optimal subsets prevent policies from exploiting lower-priority objectives at the expense of safety.

Additionally, Table I compares policies' ability to optimize the top three reward components in the preorder: safety, risk, and lane-keeping. The table reports the mean and standard deviation of cumulative rewards per episode, along with the relative percentage improvement over IQN. Results show that Pr-IQN* consistently increases rewards across all traffic densities, achieving improvements of up to 61% in safety violations, 41% in risk exposure, and 37% in lane-keeping rewards. These improvements highlight that explicitly incorporating the preorder not only enhances overall task performance but also yields safer and more reliable driving

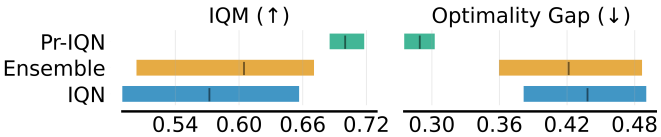


Fig. 5: Interquartile mean (IQM) and optimality gap [41], quantifying the statistical stability of a policy.

behavior by directly prioritizing high-criticality objectives.

Figures 4 and 5 complement these findings using RLiable metrics across training seeds and evaluation runs. Pr-IQN* consistently achieves the highest IQM and lowest optimality gap, confirming its reliability over IQN and Ensemble-IQN. The probability of improvement analysis further shows $P(\text{Pr-IQN} > \text{IQN})$ and $P(\text{Pr-IQN} > \text{Ensemble-IQN})$ substantially above 0.5, while the reverse probabilities remain low. These results highlight that incorporating preorder relations into distributional RL improves not only average performance but also stability and robustness across runs.

VI. Conclusion

We introduced Pr-MOMDP to encode reward preorder, proposed Quantile Dominance (QD) for distribution-aware action comparison, and developed an algorithm to extract optimal action subsets consistent with the preorder. Leveraging these, we extended IQN into Pr-IQN, where optimal subsets shape both training and decision-making. Experiments in CARLA show that Pr-IQN improves safety, success rate, and overall driving performance compared to IQN and ensemble baselines, with gains of up to 7.7%–20.3% over IQN and 11.0%–14.7% over Ensemble-IQN across traffic densities. Future work will address scalability to larger preorders and investigate how the multi-head architecture can be used for explainability and targeted fine-tuning of specific objectives. We also plan to explore hybrid setups, where certain objectives, e.g., traffic-light compliance, are evaluated by external components such as Car2X and integrated into the preorder.

Acknowledgment

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "Safe AI Engineering – Sicherheitsargumentation befähigendes AI Engineering über den gesamten Lebenszyklus einer KI-Funktion". The authors would like to thank the consortium for the successful cooperation.

TABLE I: Rewards for multiple objectives of different policies across traffic densities. $\Delta(\%)$ denotes the relative reward gain expressed as a percentage with respect to IQN. Higher values indicate better alignment of the policy with the defined objectives.

Policy	Reward components					
	Safety↑	Δ (%)	Risk↑	Δ (%)	Lane keeping↑	Δ (%)
Traffic Density 0.5						
IQN	-0.007 ± 0.003	–	-0.086 ± 0.016	–	-0.120 ± 0.013	–
Ensemble	-0.008 ± 0.006	-9.4	-0.104 ± 0.026	-20.8	-0.124 ± 0.065	-3.0
Pr-IQN*	$\mathbf{-0.004 \pm 0.002}$	$\mathbf{+42.0}$	$\mathbf{-0.051 \pm 0.011}$	$\mathbf{+40.9}$	$\mathbf{-0.076 \pm 0.013}$	$\mathbf{+36.8}$
Traffic Density 0.75						
IQN	-0.012 ± 0.006	–	-0.137 ± 0.0281	–	-0.126 ± 0.017	–
Ensemble	-0.011 ± 0.010	$+7.2$	-0.168 ± 0.042	-22.8	-0.110 ± 0.041	$+12.6$
Pr-IQN*	$\mathbf{-0.006 \pm 0.001}$	$\mathbf{+51.5}$	$\mathbf{-0.087 \pm 0.014}$	$\mathbf{+36.4}$	$\mathbf{-0.080 \pm 0.011}$	$\mathbf{+36.2}$
Traffic Density 1.0						
IQN	-0.017 ± 0.009	–	-0.164 ± 0.026	–	-0.130 ± 0.021	–
Ensemble	-0.013 ± 0.010	$+21.7$	-0.200 ± 0.038	-21.4	-0.110 ± 0.042	$+15.2$
Pr-IQN*	$\mathbf{-0.006 \pm 0.001}$	$\mathbf{+61.1}$	$\mathbf{-0.104 \pm 0.014}$	$\mathbf{+36.6}$	$\mathbf{-0.092 \pm 0.006}$	$\mathbf{+29.1}$

References

[1] A. Tampuu *et al.*, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[2] D. Coelho and M. Oliveira, "A review of end-to-end autonomous driving in urban environments," *IEEE Access*, 2022.

[3] Y. Hu *et al.*, "Planning-oriented autonomous driving," in *Conference on computer vision and pattern recognition*, 2023.

[4] Y. Lu *et al.*, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[5] B. R. Kiran *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE transactions on intelligent transportation systems*, 2021.

[6] L. Chen *et al.*, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[7] A. Abouelazm, J. Michel, and J. M. Zöllner, "A review of reward functions for reinforcement learning in the context of autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2024.

[8] W. B. Knox *et al.*, "Reward (mis) design for autonomous driving," *Artificial Intelligence*, 2023.

[9] N. Deshpande, D. Vaufreydaz, and A. Spalanzani, "Navigation in urban environments amongst pedestrians using multi-objective deep reinforcement learning," in *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[10] Z. Juozapaitis *et al.*, "Explainable reinforcement learning via reward decomposition," in *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[11] A. Censi *et al.*, "Liability, ethics, and culture-aware behavior specification using rulebooks," in *2019 international conference on robotics and automation (ICRA)*, 2019.

[12] D. Bogdoll *et al.*, "Informed reinforcement learning for situation-aware traffic rule exceptions," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[13] A. Abouelazm *et al.*, "Balancing progress and safety: A novel risk-aware objective for rl in autonomous driving," in *2025 IEEE Intelligent Vehicles Symposium (IV)*, 2025.

[14] J. Skalse *et al.*, "Defining and characterizing reward hacking," in *International Conference on Neural Information Processing Systems*, 2022.

[15] X. Han *et al.*, "AutoReward: Closed-Loop Reward Design with Large Language Models for Autonomous Driving," *IEEE Transactions on Intelligent Vehicles*, 2024.

[16] T. Joseph *et al.*, "Dream to drive: Learning conditional driving policies in imagination," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2024.

[17] H. Surmann, J. de Heuvel, and M. Bennewitz, "Multi-objective reinforcement learning for adaptable personalized autonomous driving," *arXiv:2505.05223*, 2025.

[18] H. van Seijen *et al.*, "Hybrid reward architecture for reinforcement learning," in *NeurIPS*, 2017.

[19] W. Yuan *et al.*, "Multi-reward architecture based reinforcement learning for highway driving policies," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

[20] C. Li and K. Czarnecki, "Urban Driving with Multi-Objective Deep Reinforcement Learning," in *AAMAS*, 2019.

[21] G. Jin *et al.*, "Hybrid Action Based Reinforcement Learning for Multi-Objective Compatible Autonomous Driving," *arXiv:2501.08096*, 2025.

[22] J. MacGlashan *et al.*, "Value Function Decomposition for Iterative Design of Reinforcement Learning Agents," in *NeurIPS*, 2025.

[23] A. R. M. Jamil and N. Nower, "Dynamic Weight-based Multi-Objective Reward Architecture for Adaptive Traffic Signal Control System," *International Journal of Intelligent Transportation Systems Research*, 2022.

[24] L. N. Alegre *et al.*, "AMOR: Adaptive Character Control through Multi-Objective Reinforcement Learning," in *Special Interest Group on Computer Graphics and Interactive Techniques*, 2025.

TABLE II: Evaluation metrics of different policies with various comparison metrics and thresholds across various traffic densities.

Policy	Comparison Metric	Training Preorder	Threshold ϵ	Partial Order	Evaluation metrics			
					CR ↓	OR ↓	SR ↑	RP ↑
Traffic Density 0.5								
IQN	–	–	–	–	0.248 ± 0.082	0.013 ± 0.015	0.737 ± 0.084	0.764 ± 0.053
Ensemble	–	–	–	–	0.272 ± 0.145	0.023 ± 0.022	0.704 ± 0.160	0.782 ± 0.065
MA-IQN	–	–	–	–	0.403 ± 0.039	0.002 ± 0.004	0.594 ± 0.042	0.696 ± 0.027
Pr-IQN	MV	✓	0.4	✗	0.333 ± 0.107	0.031 ± 0.015	0.635 ± 0.106	0.712 ± 0.052
Pr-IQN	CVaR	✓	0.4	✗	0.448 ± 0.057	0.032 ± 0.023	0.518 ± 0.066	0.641 ± 0.054
Pr-IQN	QD	✗	0.4	✗	0.245 ± 0.050	0.032 ± 0.014	0.722 ± 0.047	0.767 ± 0.035
Pr-IQN	QD	✓	0.4	✗	0.217 ± 0.023	0.018 ± 0.012	0.763 ± 0.029	0.789 ± 0.020
Pr-IQN	QD	✗	0.2	✗	$\mathbf{0.136 \pm 0.041}$	0.022 ± 0.011	0.816 ± 0.047	$\mathbf{0.830 \pm 0.028}$
Pr-IQN	QD	✓	0.2	✗	0.175 ± 0.029	$\mathbf{0.010 \pm 0.011}$	$\mathbf{0.816 \pm 0.030}$	0.808 ± 0.019
Pr-IQN	QD	✗	0.2	✓	0.250 ± 0.087	0.034 ± 0.018	0.692 ± 0.087	0.766 ± 0.055
Pr-IQN	QD	✓	0.2	✓	0.161 ± 0.046	0.032 ± 0.035	0.797 ± 0.087	0.805 ± 0.058
Traffic Density 0.75								
IQN	–	–	–	–	0.472 ± 0.149	0.014 ± 0.011	0.513 ± 0.156	0.611 ± 0.105
Ensemble	–	–	–	–	0.451 ± 0.171	0.016 ± 0.017	0.532 ± 0.184	0.670 ± 0.099
MA-IQN	–	–	–	–	0.617 ± 0.063	0.008 ± 0.009	0.374 ± 0.065	0.515 ± 0.045
Pr-IQN	MV	✓	0.4	✗	0.570 ± 0.142	0.030 ± 0.014	0.400 ± 0.145	0.536 ± 0.092
Pr-IQN	CVaR	✓	0.4	✗	0.692 ± 0.062	0.032 ± 0.024	0.275 ± 0.079	0.451 ± 0.073
Pr-IQN	QD	✗	0.4	✗	0.416 ± 0.058	0.015 ± 0.011	0.567 ± 0.057	0.661 ± 0.039
Pr-IQN	QD	✓	0.4	✗	0.403 ± 0.055	0.016 ± 0.012	0.580 ± 0.062	0.673 ± 0.049
Pr-IQN	QD	✗	0.2	✗	$\mathbf{0.277 \pm 0.023}$	0.032 ± 0.030	0.646 ± 0.030	0.713 ± 0.020
Pr-IQN	QD	✓	0.2	✗	0.314 ± 0.030	$\mathbf{0.005 \pm 0.004}$	$\mathbf{0.679 \pm 0.032}$	$\mathbf{0.717 \pm 0.026}$
Pr-IQN	QD	✗	0.2	✓	0.440 ± 0.138	0.028 ± 0.018	0.514 ± 0.168	0.643 ± 0.088
Pr-IQN	QD	✓	0.2	✓	0.337 ± 0.050	0.027 ± 0.023	0.630 ± 0.072	0.682 ± 0.051
Traffic Density 1.0								
IQN	–	–	–	–	0.538 ± 0.163	0.026 ± 0.034	0.434 ± 0.177	0.558 ± 0.129
Ensemble	–	–	–	–	0.485 ± 0.156	0.015 ± 0.017	0.498 ± 0.164	0.652 ± 0.091
MA-IQN	–	–	–	–	0.604 ± 0.045	0.004 ± 0.005	0.391 ± 0.048	0.532 ± 0.037
Pr-IQN	MV	✓	0.4	✗	0.616 ± 0.124	0.017 ± 0.015	0.365 ± 0.118	0.500 ± 0.075
Pr-IQN	CVaR	✓	0.4	✗	0.738 ± 0.052	0.036 ± 0.036	0.224 ± 0.063	0.413 ± 0.071
Pr-IQN	QD	✗	0.4	✗	0.507 ± 0.060	0.028 ± 0.018	0.463 ± 0.054	0.594 ± 0.045
Pr-IQN	QD	✓	0.4	✗	0.512 ± 0.044	0.032 ± 0.018	0.455 ± 0.054	0.590 ± 0.047
Pr-IQN	QD	✗	0.2	✗	$\mathbf{0.335 \pm 0.045}$	0.013 ± 0.006	0.612 ± 0.044	0.693 ± 0.028
Pr-IQN	QD	✓	0.2	✗	0.353 ± 0.045	$\mathbf{0.007 \pm 0.010}$	0.637 ± 0.043	0.688 ± 0.039
Pr-IQN	QD	✗	0.2	✓	0.453 ± 0.125	0.034 ± 0.015	0.503 ± 0.141	0.634 ± 0.075
Pr-IQN	QD	✓	0.2	✓	0.343 ± 0.036	0.014 ± 0.005	$\mathbf{0.642 \pm 0.035}$	$\mathbf{0.697 \pm 0.032}$

[25] Z. Lin *et al.*, "Distributional Reward Decomposition for Reinforcement Learning," in *NeurIPS*, 2019.

[26] P. Zhang *et al.*, "Distributional Reinforcement Learning for Multi-Dimensional Reward Functions," in *NeurIPS*, 2021.

[27] X.-Q. Cai *et al.*, "Distributional Pareto-Optimal Multi-Objective Reinforcement Learning," in *NeurIPS*, 2023.

[28] H. Wiltzer *et al.*, "Foundations of Multivariate Distributional Reinforcement Learning," in *NeurIPS*, 2024.

[29] W. Yuan *et al.*, "Multi-reward architecture based reinforcement learning for highway driving policies," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

[30] P. Halder and M. Althoff, "Sampling-based motion planning with preordered objectives," in *2025 IEEE Intelligent Vehicles Symposium (IV)*, 2025.

[31] C. Li and K. Czarnecki, "Urban driving with multi-objective deep reinforcement learning," *arXiv:1811.08586*, 2018.

[32] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv:1706.05098*, 2017.

[33] B. Helou *et al.*, "The reasonable crowd: Towards evidence-based and interpretable models of driving behavior," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[34] W. Dabney *et al.*, "Implicit quantile networks for distributional reinforcement learning," in *International conference on machine learning*, 2018.

[35] T. Théate and D. Ernst, "Risk-sensitive policy with distributional reinforcement learning," *Algorithms*, 2023.

[36] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, 1988.

[37] A. Dosovitskiy *et al.*, "Carla: An open urban driving simulator," in *Conference on robot learning*, 2017.

[38] K. Chitta *et al.*, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] M. Al-Sharman *et al.*, "Autonomous driving at unsignalized intersections: A review of decision-making challenges and reinforcement learning-based solutions," *arXiv:2409.13144*, 2024.

[40] B. Jaeger *et al.*, "Carl: Learning scalable planning policies with simple rewards," *arXiv:2504.17838*, 2025.

[41] R. Agarwal *et al.*, "Deep reinforcement learning at the edge of the statistical precipice," *NeurIPS*, 2021.