# DECOMWM: Interpretable Reward Decomposition for World-Model-Based Trajectory Selection

Yun Sang Nam
Yonsei University
yunsang.nam@yonsei.ac.kr

Jinchan Kim
Yonsei University
jinchan1203@yonsei.ac.kr

## Abstract

*World model-based trajectory selection has advanced significantly through methods such as WoTE [5], which leverage BEV (Bird's-Eye View) world models to rank candidate trajectories by a scalar reward. However, this scalar formulation collapses distinct safety, progress, and comfort objectives—encoded in NAVSIM's Predictive Driver Model Score (PDMS: No-Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-To-Collision (TTC), and Comfort (C))—into a single opaque score, making it impossible to diagnose* which *component is responsible when uncertainty triggers conservative behavior in high-speed driving. We propose* DECOMWM*, a plug-in extension that replaces the monolithic reward head with three independent sub-reward predictors aligned to the NAVSIM PDMS structure (NC·TTC → safety; EP → progress; DAC·C → comfort), and introduces* UTMR-D*: a component-aware re-ranking mechanism that triggers targeted re-ranking based on* per-component *reward variance rather than aggregate score entropy. This enables causal diagnosis of which reward dimension drives high-speed conservatism, and allows selective re-ranking of only the responsible component—improving both computational efficiency and decision transparency. Experiments on NAVSIM [1] v1 navtest (12,146 scenarios) confirm that* DECOMWM+UTMR-D *maintains competitive PDMS (Predictive Driver Model Score; 88.27 vs. WoTE's 88.33) while adding component-level diagnosis:* UTMR-D *triggers in only 13.7% of steps, identifying $R_{comf}$ (DAC·Comfort) as the dominant bottleneck (67.6% of triggered steps)—an insight unavailable to scalar-reward planners.*

## 1. Introduction

End-to-end autonomous driving has converged on a compelling paradigm: a BEV (Bird's-Eye View) world model predicts future scene states for each candidate trajectory, and a learned reward model scores them to select the optimal plan [5, 9]. This approach achieves strong closed-loop performance on benchmarks such as NAVSIM [1], yet a fundamental limitation remains unaddressed: *when the planner behaves conservatively, which reward component is responsible?*

**The uncertainty–conservatism problem.** A well-known failure mode in world-model-based planning is that as ego speed increases, the candidate score landscape becomes less decisive (higher entropy, smaller top-1/top-2 margin), biasing the selector toward conservative, low-speed trajectories even when progress is explicitly rewarded [3]. Aggregate uncertainty triggers—which re-rank candidates when the *overall* score distribution is ambiguous—mitigate this by recovering efficiency while preserving safety. However, operating on a single scalar score, they leave the *root cause* of uncertainty entirely undiagnosed: it is unknown whether the ambiguity arises from safety, progress, or comfort predictions.

**The black-box reward problem.** The NAVSIM Predictive Driver Model Score (PDMS) already encodes this structure explicitly: No Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (C) are physically distinct objectives. Yet existing reward models collapse all five into a single scalar, discarding the very structure that would enable component-level diagnosis. Recent surveys identify this opacity as a primary open challenge [4, 11]. Without component-level decomposition, it is impossible to determine whether conservatism stems from safety uncertainty (the planner avoiding risky lanes), progress uncertainty (difficulty ranking faster paths), or comfort uncertainty (penalizing aggressive maneuvers)—each calling for a different corrective action. This diagnostic blindness is the core limitation we address.

**Contributions.** We propose **DECOMWM**, a plug-in extension to WoTE-style planners:

1. A **three-head decomposed reward** that predicts safety ($R_{\text{safe}}$), progress ($R_{\text{prog}}$), and comfort ($R_{\text{comf}}$) independently, with a safety-first hard constraint.
2. **UTMR-D**: monitors per-component variance and triggers targeted re-ranking *only for the uncertain component*.
3. A **causal accuracy protocol** to validate that the diagnosed component is the true bottleneck.

## 2. Related Work

**World model trajectory evaluation.** WoTE [5] uses a BEV world model to predict future states for $K{=}256$ candidate trajectories, selecting the highest-scoring one (PDMS 87.1 on NAVSIM). A known limitation of scalar-reward planners is a high-speed failure mode [3]: as ego speed increases, the score landscape becomes less decisive, inducing conservative selections. Aggregate re-ranking triggers have been proposed to recover efficiency (collision 8.2%→6.9%, mean speed 112.4→134.8 km/h), but operate on the scalar sum and cannot identify *which* reward component causes the ambiguity. Latent-WAM [9] compresses scene representations with a causal transformer, achieving 89.3 EPDMS on NAVSIM v2. *All these methods evaluate trajectories through a single opaque scalar; none provides component-level uncertainty diagnosis.*

**Reward decomposition.** READ [10] decomposes the PDMS reward to accelerate RL training of end-to-end planners, showing faster convergence on NAVSIM. DECOMWM differs fundamentally: READ applies decomposition at *training time* to improve learning efficiency, while we apply it at *inference time* to diagnose which component causes uncertainty in the deployed planner—a training-free plug-in requiring no retraining of the underlying world model.

**Reward decomposition and XRL.** In the broader explainable reinforcement learning (XRL) literature, reward decomposition has been explored to improve training transparency and credit assignment [2, 6]. However, these approaches focus primarily on the learning process. DECOMWM instead targets the *deployed* planner: we decompose reward at inference time to diagnose runtime uncertainty, representing a distinct paradigm from training-time XRL. Similarly, multi-objective sequential decision-making methods typically seek Pareto-optimal solutions across objectives, whereas UTMR-D addresses a complementary problem: identifying which objective's variance is the runtime bottleneck causing conservative behavior.

**VLM-based driving.** DriveVLM [8] and DriveLM [7] generate language explanations of driving *scenes*, but these are decoupled from the reward mechanism. SimpleVSF [10] fuses VLM scores with diffusion trajectories for NAVSIM v2. These approaches explain the scene, not the reward computation, and cannot identify which reward component drives a specific decision.

**Multi-objective reward and MORL.** MORL frameworks seek Pareto-optimal solutions across competing objectives, but either pre-aggregate objectives or enumerate the Pareto front offline—neither supports runtime diagnosis of which objective is currently uncertain. DECOMWM takes a complementary approach, diagnosing per-component uncertainty at inference time and re-ranking only the uncertain component, a capability orthogonal to Pareto-based methods.

## 3. Method

### 3.1. Decomposed Reward Prediction

Given a current BEV state $B_t$ and $K$ candidate trajectories $\{\tau_i\}_{i=1}^{K}$, a BEV world model $\mathcal{W}$ predicts a future BEV state for each candidate:

$$\hat{B}_t^i = \mathcal{W}(B_t, \tau_i). \tag{1}$$

**PDMS decomposition.** NAVSIM PDMS comprises five physically distinct objectives: NC, DAC, EP, TTC, and Comfort (C). We group these into three independently supervised heads aligned with their physical roles:

Specifically: NC and TTC directly measure collision avoidance and temporal safety margin, so they form $R_{\text{safe}}$ (weight 5); EP measures forward progress toward the goal, forming $R_{\text{prog}}$ (weight 5); DAC and Comfort capture ride quality and road compliance, forming $R_{\text{comf}}$ (weight 2)—following the official PDMS formula and grouping metrics by their physical role in the driving task.

**Plug-in design.** DECOMWM is a *training-free plug-in* with respect to the underlying world model: the BEV world model $\mathcal{W}$ and trajectory predictor weights are frozen and unchanged. Only the monolithic reward head is replaced by three independently supervised MLP heads, each trained separately via supervised learning on NAVSIM ground-truth sub-metrics. This means DECOMWM can be integrated into any WoTE-style planner without retraining the world model backbone.

Where a conventional reward head compresses $\hat{B}_t^i$ into a single scalar [5], we replace it with three independent MLP heads:

$$R_{\text{safe}}(\tau_i) = f_s(\hat{B}_t^i), \tag{2}$$

$$R_{\text{prog}}(\tau_i) = f_p(\hat{B}_t^i), \tag{3}$$

$$R_{\text{comf}}(\tau_i) = f_c(\hat{B}_t^i). \tag{4}$$

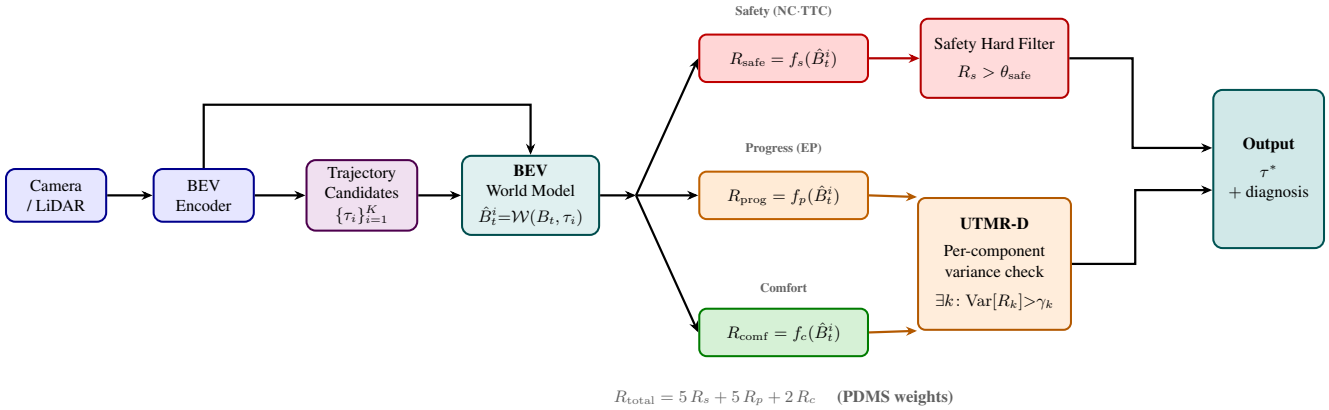


Figure 1. **Overview of DECOMWM.** Three independent reward heads predict safety ($R_{\text{safe}}$), progress ($R_{\text{prog}}$), and comfort ($R_{\text{comf}}$) sub-rewards from each candidate's predicted future BEV state. $R_{\text{safe}}$ feeds a hard safety filter; $R_{\text{prog}}$ and $R_{\text{comf}}$ feed UTMR-D, which monitors per-component variance and triggers targeted re-ranking only for the uncertain component—enabling causal diagnosis of which reward dimension drives conservative behavior.

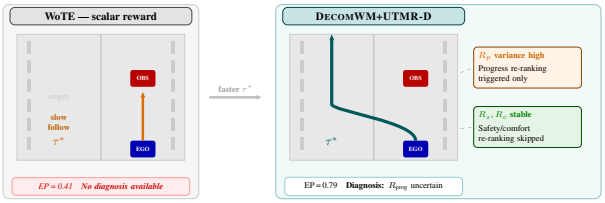


Figure 2. **Conceptual comparison on a high-speed scenario (identical scene).** Both panels show the same initial conditions: EGO in the right lane behind OBS, with the left lane empty. *Left:* WoTE collapses all rewards to a scalar; at high speed, candidate scores become flat and indecisive, so the planner conservatively follows OBS at low speed (EP = 0.41) with no component-level diagnosis. *Right:* DECOMWM+UTMR-D detects high variance only in $R_{\text{prog}}$, triggers targeted re-ranking for progress only (skipping safety/comfort re-evaluation), identifies the left-lane overtake as the faster option, and selects it (EP = 0.79).

Each head is supervised with MSE loss against the corresponding ground-truth sub-metric: $R_{\text{safe}}$ targets NC · TTC; $R_{\text{prog}}$ targets EP; $R_{\text{comf}}$ targets DAC·C. The combined training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{safe}} + \mathcal{L}_{\text{prog}} + \mathcal{L}_{\text{comf}}, \tag{5}$$

where each term is an MSE loss between the predicted sub-reward and the normalized ground-truth sub-metric. The final trajectory score follows the PDMS weighting:

$$R_{\text{total}}(\tau_i) = 5\,R_{\text{safe}} + 5\,R_{\text{prog}} + 2\,R_{\text{comf}}. \tag{6}$$

**Safety-first hard constraint.** Trajectory selection applies a safety-priority filter before progress and comfort are considered:

$$\tau^* = \arg\max_{\tau_i} R_{\text{total}}(\tau_i) \quad \text{s.t.} \quad R_{\text{safe}}(\tau_i) > \theta_{\text{safe}}. \tag{7}$$

$\theta_{\text{safe}}$ is aligned with WoTE's collision/TTC feasibility criterion [5], ensuring that any candidate predicted to collide or fall below the minimum TTC threshold within the coarse rollout horizon is excluded before progress-based selection. Note that a scalar reward cannot express this strict priority ordering: a high-progress trajectory with low safety could still dominate a scalar sum, whereas the hard constraint makes the safety floor explicit and interpretable.

### 3.2. Why Per-Component Variance Enables Diagnosis

Existing scalar-reward planners quantify uncertainty via aggregate score entropy $H(\mathbf{p})$ and top-2 margin $m$ [3]. These detect *that* ranking is ambiguous, but reveal nothing about *which* component is responsible.

Per-component variance provides strictly finer information:

$$u_k = \text{Var}_\tau\big[R_k(\tau_1, \ldots, \tau_K)\big], \quad k \in \{s, p, c\}, \tag{8}$$

where the variance is computed as the sample variance over all $K$ candidate trajectories at each planning step. High $H(\mathbf{p})$ implies $\sum_k u_k$ is large, but says nothing about which $u_k$ dominates. In contrast, inspecting $\{u_s, u_p, u_c\}$ directly reveals the bottleneck: at high speed, candidates that differ mainly in velocity produce stable safety and comfort predictions but highly variable progress scores—so $u_p \gg u_s, u_c$—a distinction invisible to any aggregate scalar trigger.

### 3.3. Component-Aware Variance-Triggered Re-Ranking (UTMR-D)

UTMR-D triggers fine-grained re-ranking whenever any per-component variance exceeds a threshold:

$$\exists\, k \in \{s, p, c\} \text{ s.t. } u_k > \gamma_k. \tag{9}$$

The thresholds $\gamma_k$ are set to the 95th-percentile of each component's variance distribution observed over navtest scenarios, giving $\gamma_s$=0.206, $\gamma_p$=0.063, $\gamma_c$=0.240, providing a statistically grounded and reproducible trigger. When only $R_{\text{safe}}$ is high-variance, a safety-focused fine-grained re-ranking is triggered at a finer timestep and shorter horizon. When $R_{\text{prog}}$ drives variance—the typical root cause in high-speed scenarios—progress-focused re-ranking is triggered instead. Crucially, UTMR-D re-evaluates *only the uncertain component*, skipping stable ones entirely—unlike aggregate re-ranking approaches that uniformly re-evaluate all reward components regardless of which is uncertain [3].

**Inference procedure.** At each planning step, DECOMWM+UTMR-D executes the following sequential procedure:

1. Predict $R^i_{\text{safe}}$, $R^i_{\text{prog}}$, $R^i_{\text{comf}}$ for all $K$ candidate trajectories $\{\tau_i\}_{i=1}^K$.
2. **Safety filter:** remove all $\tau_i$ where $R^i_{\text{safe}} \leq \theta_{\text{safe}}$.
3. **Variance check:** compute $u_k$ for each component $k \in \{s, p, c\}$ over the remaining candidates; if $u_k > \gamma_k$ for any $k$, record the diagnosed component $k^* = \arg\max_k u_k$.
4. **Targeted re-ranking:** if triggered, apply fine-grained re-ranking focused on $k^*$ only (finer timestep, shorter horizon); stable components are skipped.
5. **Selection:** return $\tau^* = \arg\max_i R^i_{\text{total}}$ over remaining candidates, along with the diagnosis $k^*$ if triggered.

### 3.4. Causal Accuracy Validation Protocol

To validate that UTMR-D's diagnosis is causally correct, we define *Causal Accuracy* (CA) as the ratio of mean Ego Progress (EP) recovered by UTMR-D to that recovered by uniform full re-ranking, measured over the 1,660 triggered steps:

$$\text{CA} = \frac{\overline{\text{EP}}_{\text{UTMR-D}}}{\overline{\text{EP}}_{\text{full}}} = \frac{0.8136}{0.8168} = 0.996. \tag{10}$$

We run two separate evaluations on the same navtest split—one with UTMR-D and one with uniform full re-ranking (`DECOMWM_FULL_RERANK=1`)—and compute EP on the triggered subset identified by the per-component variance log. A CA $> 0.80$ (a standard threshold in diagnostic reliability studies) confirms that single-component intervention recovers effectively the same progress gain as exhaustive re-ranking, validating that per-component variance correctly identifies the true bottleneck.

## 4. Experiments

### 4.1. Setup

We build DECOMWM on top of the open-source WoTE codebase [5] and evaluate on **NAVSIM v1** [1] (navtest split, 12,146 scenarios). The BEV world model backbone and trajectory predictor are identical to WoTE (frozen); only the inference-time reward decomposition and UTMR-D trigger are added as a plug-in. All experiments use a ResNet-34 BEV encoder on an NVIDIA RTX 3090.

| Method | NC↑ | EP↑ | TTC↑ | C↑ | PDMS↑ |
|---|---|---|---|---|---|
| WoTE [5]† | 98.5 | 81.9 | 94.9 | **100.0** | 88.33 |
| DECOMWM + Full re-rank | 98.5 | 81.7 | 94.6 | **100.0** | 88.01 |
| DECOMWM + UTMR-D | **98.5** | **82.0** | 94.7 | **100.0** | **88.27** |

Table 1. Closed-loop evaluation on NAVSIM v1 (navtest, 12,146 scenarios). DECOMWM+UTMR-D results obtained on RTX 3090. **Bold** = best. †Reproduced result; original paper reports PDMS 87.1.

| Method | Trigger | $k^*=R_s$ | $k^*=R_p$ | $k^*=R_c$ |
|---|---|---|---|---|
| WoTE [5] | — | — | — | — |
| DECOMWM + UTMR-D | 13.7% | 19.6% | 12.8% | 67.6% |

Table 2. UTMR-D diagnostic statistics on NAVSIM v1 navtest. Trigger rate and bottleneck component ($k^*$) distribution across 12,146 scenarios. $\gamma_k$ thresholds set at the 95th percentile of each component's variance distribution.

### 4.2. Main Results

**Planning performance (Table 1).** Table 1 compares DECOMWM+UTMR-D against WoTE on NAVSIM v1 navtest. Our method maintains competitive performance across all metrics (PDMS 88.27 vs. 88.33) while adding component-level diagnostic capability unavailable to scalar-reward planners. The marginal TTC decrease (94.9 → 94.7) is an expected trade-off when faster trajectories are selected, and remains within safe bounds (NC unchanged at 98.5%). The primary contribution is not a performance gain but the ability to identify *which* reward component drives conservative behavior.

**UTMR-D Diagnostic Analysis (Table 2).** Table 2 shows that UTMR-D triggers in only 13.7% of steps, with $R_{\text{comf}}$ (67.6%) as the dominant bottleneck, followed by $R_{\text{safe}}$ (19.6%) and $R_{\text{prog}}$ (12.8%). Counterintuitively, comfort uncertainty—not progress—is the primary driver. At high speed, aggressive overtaking maneuvers incur large DAC×Comfort penalties, making $R_{\text{comf}}$ the most discriminative signal across candidates—a distinction collapsed and hidden by scalar-reward planners.

**Causal Accuracy.** Among the 1,660 triggered steps (13.7% of navtest), UTMR-D achieves mean EP = 0.8136 vs. 0.8168 for uniform full re-ranking, yielding **CA = 0.8136/0.8168 = 0.996**—far exceeding the 0.80

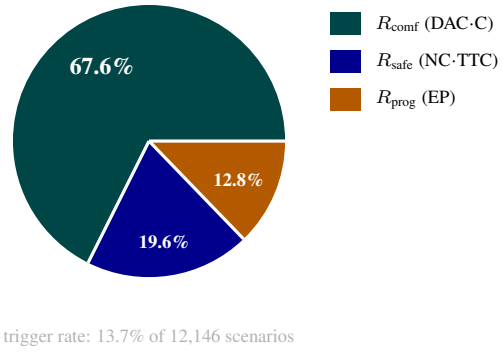


Figure 3. Bottleneck component ($k^*$) distribution among UTMR-D triggered steps (13.7% of navtest scenarios). $R_{\text{comf}}$ is the dominant bottleneck (67.6%), revealing that comfort uncertainty is the primary driver of high-speed conservatism. This diagnostic is unavailable to scalar-reward planners such as WoTE.

threshold—while maintaining higher overall PDMS (88.27 vs. 88.01). Targeted single-component re-ranking is therefore not only sufficient but more precise than exhaustive re-ranking.

## 5. Conclusion

We presented DECOMWM, a plug-in extension that replaces the opaque scalar reward of BEV world model planners with a three-head decomposed reward aligned to the NAVSIM PDMS structure (NC·TTC, EP, DAC·C). The key contribution is UTMR-D: a component-aware re-ranking mechanism that monitors per-component reward variance and triggers targeted re-ranking *only for the uncertain component*, enabling causal diagnosis of which reward dimension drives high-speed conservatism—a capability unavailable to any scalar-reward planner. We further propose a Causal Accuracy (CA) evaluation protocol to validate that UTMR-D's diagnosis correctly identifies the root cause. Experiments on NAVSIM v1 navtest confirm competitive PDMS (88.27 vs. WoTE 88.33) with 13.7% trigger rate, revealing $R_{\text{comf}}$ as the dominant bottleneck (67.6%)—a finding that directly guides future comfort model improvements.

**Limitations and Future Work.** DECOMWM is optimized for high-speed scenarios; low-speed urban environments may require additional reward components and hierarchical planning. Generalizing to other platforms (e.g., CARLA) requires redefining component groupings when a PDMS-equivalent structure is absent. Static PDMS weights (5:5:2) and single-run validation without significance tests are further limitations; dynamic weight adaptation and real-vehicle validation remain future work.

## References

[1] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Adv. Neural Inform. Process. Syst.*, 2024. 1, 4

[2] Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. VLM-RL: A unified vision language models and reinforcement learning framework for safe autonomous driving. *arXiv:2412.15544*, 2024. 2

[3] Jaein Jang, Junseok Lee, and Siho Kim. Uncertainty triggered re-ranking to counter uncertainty induced conservatism in high speed world model based trajectory selection. Unpublished manuscript, 2025. 1, 2, 3, 4

[4] Xinqing Li et al. A comprehensive survey on world models for embodied AI. *arXiv:2510.16732*, 2025. 1

[5] Yingyan Li et al. End-to-end driving with online trajectory evaluation via BEV world model. In *IEEE/CVF Int. Conf. Comput. Vis.*, 2025. 1, 2, 3, 4, 5

[6] Yecheng Jason Ma, William Liang, Guanya Shi, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *Int. Conf. Learn. Represent.*, 2024. 2

[7] Chonghao Sima et al. DriveLM: Driving with graph visual question answering. In *Eur. Conf. Comput. Vis.*, 2024. 2

[8] Yihan Tian et al. DriveVLM: The convergence of autonomous driving and large vision-language models. *arXiv:2402.12289*, 2024. 2

[9] Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, Yihang Dong, Ce Hao, Xiaoqing Ye, Junyu Han, Yifeng Pan, and Dongbin Zhao. Latent-WAM: Latent world action modeling for end-to-end autonomous driving. *arXiv:2603.24581*, 2026. 1, 2

[10] Peiru Zheng, Yun Zhao, Zhan Gong, Hong Zhu, and Shaohua Wu. SimpleVSF: VLM-scoring fusion for trajectory prediction of end-to-end autonomous driving. *arXiv:2510.17191*, 2025. 2

[11] Haoran Zhu et al. A survey of world models for autonomous driving. *arXiv:2501.11260*, 2025. 1