GRADE: Guiding Realistic Autonomous Driving with Adaptive Trajectory Evolution

Supplementary Material

1. Mathematical Proof of Adaptive Evolution

We propose a weakness-aware evolution method that (i) precisely identifies scene-specific bottleneck, (ii) quantifies the optimization potential of each metric, and (iii) adaptively adjusts the corresponding weights. To quantify the expected improvement potential of each dimension, we model the importance of elite trajectories via the initial scores:

$$q_j = \frac{\exp\big(R(x_j, \boldsymbol{w}^0) \cdot \tau\big)}{\sum_{k=1}^{M} \exp(R(x_k, \boldsymbol{w}^0) \cdot \tau)}, \tag{1}$$

where x_j represents the j-th trajectory among the top-M trajectories, $\boldsymbol{w}^0$ denotes the initial weight vector before adaptive fluctuation, $R(\cdot)$ is the reward function, and τ is the tunable temperature. Then the weakness factor of each metric is then modeled as follows:

$$\delta_i = \sum_{j=1}^{M} q_j \cdot \mathbb{1}[s_i^j < s_i^{\text{ref}}], \tag{2}$$

where $\mathbb{1}[s_i^j < s_i^{\text{ref}}]$ is an indicator function that equals 1 only if the score s_i^j of the j-th trajectory on the i-th metric is inferior to the reference value s_i^{ref}, and 0 otherwise. The factor $\delta_i \in [0, 1]$ measures the residual optimization space of metric i. When $\delta_i = 0$, the population has already approached the best attainable value for that metric, rendering further optimization unnecessary. Conversely, $\delta_i = 1$ indicates that the elite trajectories selected by the original weights neglected this metric, mandating its improvement in subsequent evolution. Within the limited iteration rounds, the model can concentrate on those metrics that truly critical and merit refinement in the present scenario, thereby boosting optimization efficiency. After computing the weakness factors, the weights are adjusted according to

$$\bar{w}_i = \frac{(1 + \delta_i) \cdot w_i^0}{\sum_{k=1}^{N_w} (1 + \delta_k) \cdot w_k^0}, \tag{3}$$

where w_i^0 is the initial weight of the i-th metric while $\bar{w}_i$ denotes the fluctuated weight. Metrics achieving maximal δ_i become priority optimization targets, as they present the greatest performance potential. Our weakness-aware evolution method demonstrably satisfies this requirement, as formally verified as follows:

Theorem 1 (Bottleneck Prioritization)**.** *When the weakness factor δ_i attains the maximum value among the set $\mathcal{M}_r$ of metrics composing the reward function ($\delta_i = \max_{j \in \mathcal{M}_r} \delta_j$), the updated weight w_i is guaranteed to satisfy:*

$$\bar{w}_i = \frac{(1 + \delta_i) \cdot w_i^0}{\sum_{j \in \mathcal{M}_r} (1 + \delta_j) \cdot w_j^0} \geq w_i^0$$

Proof. Since $\delta_j \leq \delta_i$ for all $j \in \mathcal{M}_r$ (by maximality condition),

$$\sum_{j \in \mathcal{M}_r} (1 + \delta_j) w_j^0 \leq \sum_{j \in \mathcal{M}_r} (1 + \delta_i) w_j^0$$

Thus:

$$\bar{w}_i = \frac{(1 + \delta_i) \cdot w_i^0}{\sum_{j \in \mathcal{M}_r} (1 + \delta_j) \cdot w_j^0} \geq \frac{(1 + \delta_i) w_i^0}{(1 + \delta_i) \sum\limits_{j \in \mathcal{M}_r} w_j^0}$$

Apply the weight normalization:

$$\sum_{j \in \mathcal{M}_r} w_j^0 = 1$$

Therefore:

$$\bar{w}_i = \frac{(1 + \delta_i) \cdot w_i^0}{\sum_{j \in \mathcal{M}_r} (1 + \delta_j) \cdot w_j^0} \geq \frac{(1 + \delta_i) w_i^0}{(1 + \delta_i) \sum\limits_{j \in \mathcal{M}_r} w_j^0} = w_i^0$$

The equality condition $\bar{w}_i = w_i^0$ holds if and only if:

$$\sum_{j \in \mathcal{M}_r} (1 + \delta_j) w_j^0 = \sum_{j \in \mathcal{M}_r} (1 + \delta_i) w_j^0$$

which requires $\delta_j = \delta_i$ for all $j \in \mathcal{M}_r$, in other words, when all metrics have identical weakness factors. □

Define the relative change rate $\mu = \frac{\bar{w} - w^0}{w^0}$ as the relative deviation of the dynamic weight for each metric. It can be proven that the adjustment is both bounded and stable, with its magnitude fully governed by the initial weights. As illustrated in Figure 6, this property guarantees that the direction of iterative optimization after the dynamic weight fluctuation remains guided by the initial weights.

Theorem 2 (Boundedness)**.** *The dynamically adjusted weights satisfy $0.5w^0 < \bar{w} < 2w^0$, with boundary values achieved under extreme weakness factor configurations.*

Proof. For any metric $i \in \mathcal{M}_r$, when its weakness factor $\delta_i = 1$ and all other factors $\delta_{\neg i} = 0$, the adjusted weight $\bar{w}_i$ attains its maximum value:

$$\bar{w}_i^{\max} = \frac{(1 + \delta_i) \cdot w_i^0}{\sum_{j \in \mathcal{M}_r} (1 + \delta_j) \cdot w_j^0} = \frac{2w_i^0}{2w_i^0 + \sum_{\neg i} w_{\neg i}^0} = \frac{2w_i^0}{1 + w_i^0}$$

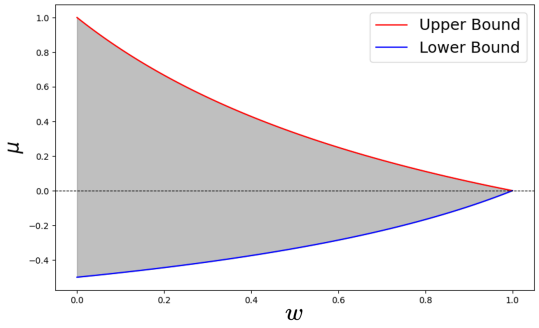

Figure 6. The rate of change μ vs the initial weight w

Here the normalization condition $\sum_{\neg i} w^0_{\neg i} = 1 - w^0_i$ is applied. Likewise, when $\delta_i = 0$ and all other factors $\delta_{\neg i} = 1$, the minimum value of $\bar{w}_i$ is:

$$\bar{w}_i^{\min} = \frac{(1+\delta_i)\cdot w_i^0}{\sum_{j\in\mathcal{M}_r}(1+\delta_j)\cdot w_j^0} = \frac{w_i^0}{w_i^0 + \sum_{\neg i} 2w^0_{\neg i}} = \frac{w_i^0}{2 - w_i^0}$$

Since $0 < w_i^0 < 1$:

$$0.5w_i^0 < \frac{w_i^0}{2-w_i^0} \le \bar{w}_i \le \frac{2w_i^0}{1+w_i^0} < 2w_i^0$$

$$-0.5 < \mu_i < 1$$

Given that the proposed method confines weight fluctuations to a bounded range, an optional scaling factor can be introduced for further fine-tuning. □

Theorem 3 (Stability)**.** *The fluctuation range* $\Delta\mu_i = (w_i^{max} - w_i^{min})/w_i^0$ *decays monotonically as* w_i^0 *increases, ensuring optimization stability.*

Proof. The fluctuation range is defined as:

$$\Delta\mu_i(w_i^0) = \frac{(w_i^{\max} - w_i^{\min})}{w_i^0} = \frac{2}{1+w_i^0} - \frac{1}{2-w_i^0},$$

where $0 < w_i^0 < 1$.To analyze the stability property, we compute the derivative of $\Delta\mu_i$ with respect to w_i^0:

$$\frac{d\Delta\mu_i(w_i^0)}{dw_i^0} = -\frac{2}{(1+w_i^0)^2} - \frac{1}{(2-w_i^0)^2} < 0$$

This negative derivative guarantees bounded fluctuations, thereby shielding evolution from destabilizing weight swings. □

2. Evaluation metrics of NuPlan

The nuPlan closed-loop evaluation metrics contain many driving factors, as shown in Table 6. Each metric corresponds to several conditions and thresholds, and if during the driving case some of them are broken, score punishment would be given. The final score is the weighted summation of metrics revised by 0-1 multiplier to ensure the priority of safety. The detailed rules are described in the official website of nuPlan challenge *https://nuplan-devkit.readthedocs.io/en/latest/metrics_description.html*.

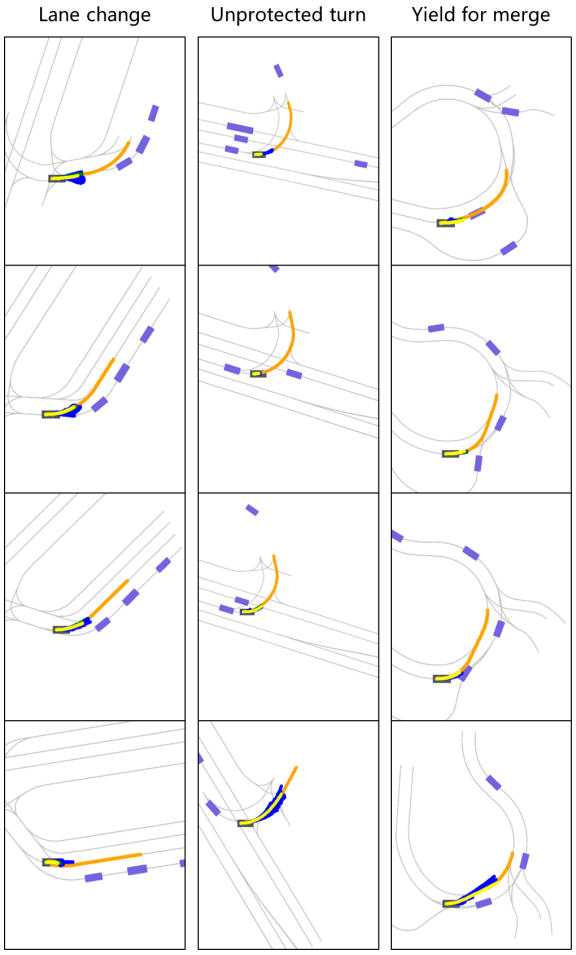

Figure 7. More nuPlan closed-loop cases

3. More Typical Cases of Simulation

As shown in Table 5, enhancing PDM-Closed with GRADE improves performance across most metrics, particularly in hard scenarios, validating the effectiveness of our design. More typical interactive driving cases in nuPlan closed-loop experiments are shown in Figure 7, where the orange line is the human trajectory of future 8 seconds, blue lines are the derived trajectory clusters of Diffusion-WAE, the yellow line is the best result among the cluster, and grey lines are road lane centerlines, and purple boxes are other agents.

4. Settings of Real-vehicle Validations

The real-vehicle experiment is conducted in the Connected and Automated Vehicle (CAV) Demonstration District,

Table 5. Comparison results on the nuPlan benchmark. **Bold** indicates the best performance. Our method achieves significant improvements on Test14 and Test14-hard split.

Method	Val14		Test14-hard		Test14	
	NR	R	NR	R	NR	R
PDM-Closed	**92.84**	**92.12**	65.08	75.19	90.05	91.63
PDM-Closed + GRADE	91.19	91.26	**67.49**	**76.62**	**91.00**	**91.92**

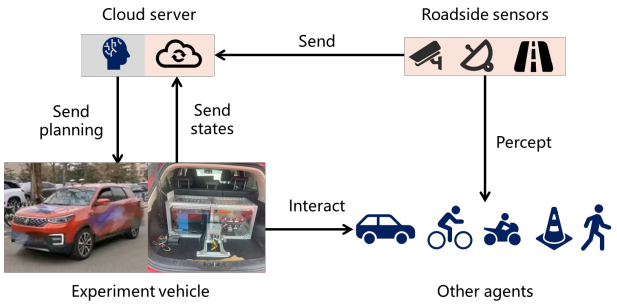

Figure 8. The vehicle-road-cloud structure of real-road experiment platform

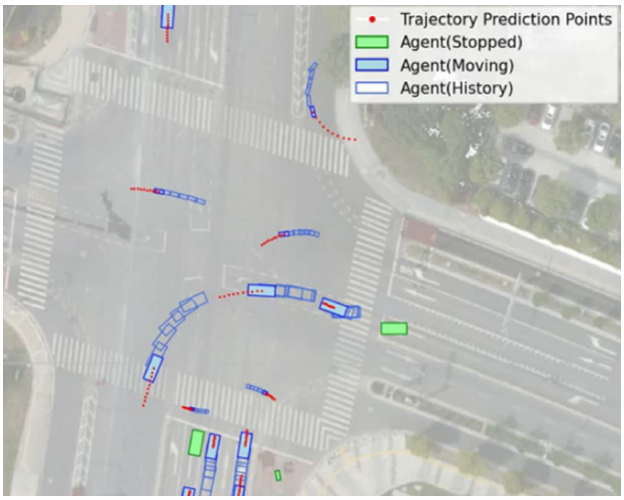

Figure 9. An visualization of the roadside perception, prediction, and planning

Suzhou, China which is established based on the vehicle-road-cloud system, as shown in Figure 8. All perceptions are finished by roadside sensors equipped along the lanes, and the perception results, roadmap information, and ego vehicle states are transmitted to the cloud server. After gathering all needed inputs, the cloud server run the planning algorithms and sends the best trajectory to the vehicle. Finally, the vehicle conducts a trajectory-following task and interacts with other agents. The perception effect of the roadside sensors is illustrated in Figure 9.

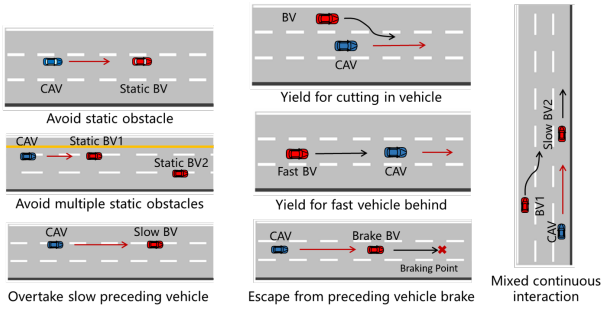

Figure 10. Experiment scenes of real-vehicle validations

5. Full Case of Real-vehicle validations

In the real-vehicle validations, we tested our proposed algorithm on six interaction scenes, as demonstrated in Figure 10 where the blue one is the controlled connected and automated vehicle (CAV) and the red ones are human-driven background vehicle (BV). These scenes contain common longitudinal and lateral interaction types such as overtaking, yielding, cutting in, etc.

The validation results are shown in Figure 11. To be noted that since the experiment is based on the vehicle-road-cloud platform, The communication process introduces approximately 2 seconds of time delay, resulting in vehicle behavior lagging behind in certain scenarios although the real-time decision-making outcomes are reasonable. It should be emphasized that the proposed algorithm itself does not exhibit significant processing delays. Consequently, even under prolonged communication delay conditions, the experiment ego vehicle successfully passed complex interactions. The experimental results demonstrate that the proposed planning method possesses practical applicability in real-world road scenarios. When confronted with continuously complex and coupled traffic situations, the ego vehicle can execute reasonable decision-making and control, employing a unified decision-making paradigm to perform diverse behaviors such as car-following, lane-changing, overtaking, and collision avoidance, thereby achieving predefined driving objectives with superior performance compared to baseline models.

Figure 11. Experiment results of real-vehicle validations

Metric	Conditions and Thresholds
No at-fault collision	collision-with-vru = 1 collision-with-vehicle = 1 collision-with-object = 1
Drivable Area Compliance	max-violation-threshold = 0.3 m
Driving Direction Compliance	driving-direction-violation-threshold = 2 m driving-direction-violation-threshold = 6 m time-horizon = 1 s
Making progress	min-progress-threshold = 0.2
Time to Collision	time-step-size = 0.1s time-horizon = 3.0s least-min-ttc = 0.95s
Speed limit compliance Ego progress along the expert's route ratio	max-complicane = 1.0 max-overspeed = 2.23 m/s score-progress-threshold = 0.1 m
Comfort Displacement error	min-lon-accel = -4.05 m/s^2 max-lon-accel = 2.40 m/s^2 max-abs-lat-accel = 4.89 m/s^2 max-abs-yaw-accel = 1.93 rad/s^2 max-abs-yaw-rate = 0.95 rad/s max-abs-lon-jerk = 4.13 m/s^3 max-abs-mag-jerk = 8.37 m/s^3 discount-factor = 1
Displacement error with yaw	discount-factor = 1 heading-diff-weight = 2.5

Table 6. NuPlan closed-loop evaluation metrics