HYPERPARAMETER SENSITIVITY ANALYSIS OF REINFORCEMENT LEARNING IN AUTONOMOUS DRIVING ENVIRONMENTS

Marihan Shehata, Mohammed Moness, and Ahmed M. Mostafa

Faculty of Engineering, Minia University, Minia, Egypt

ABSTRACT

Hyperparameter tuning plays a critical role in reinforcement learning (RL), particularly in safety-critical domains such as autonomous driving. In this work, we conduct a large-scale empirical analysis of hyperparameter sensitivity for two of the most widely used RL — Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) — using theCommonRoad-RL framework and the highD dataset. Functional analysis of variance (FANOVA) is employed to quantify main and interaction effects. Results show that performance variation in both algorithms is dominated by hyperparameter interactions, accounting for over 90% in PPO and nearly 88% in SAC, contrasting prior findings in simpler RL benchmarks. PPO is most sensitive to value learning and gradient stability, whereas SAC is driven by replay and training parameters. These findings highlight the need for interaction-aware tuning strategies to ensure robust RL deployment in complex driving tasks.

KEYWORDS

Autonomous driving, Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC) Hyperparameter optimization, Hyperparameter sensitivity

1. INTRODUCTION

Reinforcement learning (RL) has demonstrated remarkable potential in tackling complex decision-making problems, such as autonomous driving[1-3]. However, the performance of RL algorithms is highly sensitive to hyperparameter choices, with even seemingly minor changes often leading to drastic differences in learning stability and final performance[4-6].This issue is especially pronounced in real-world, safety-critical domains such as autonomous driving, where robust and efficient learning is essential.

While extensive work has analysed hyperparameter sensitivity in synthetic or simulated environments[7,8],a notable gap remains in systematic studies of RL for autonomous driving. In such settings, the challenges of high-dimensional sensory input and sparse rewards can amplify the effects of hyperparameter choices. Despite progress in automated reinforcement learning (AutoRL) and hyperparameter optimization (HPO) techniques [9-13], These tools have seen limited adoption in the autonomous driving domain, and practitioners often rely on default settings or manual tuning.

In this work, we present a comprehensive empirical study of hyperparameter sensitivity for two widely used continuous-action reinforcement learning algorithms—Proximal Policy Optimization (PPO)[14], a stable and simple on-policy method, and Soft Actor-Critic (SAC)[15], an off-policy

David C. Wyld et al. (Eds): NATL, MLTEC, SIGEM, SEAPP, CSEA, FUZZY, DMDBS – 2025
pp. 53-63, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.152205

algorithm known for sample efficiency and robust exploration. We use the CommonRoad-RL framework [16], a benchmark for motion planning in complex driving scenarios, and the highD dataset[17], a real-world highway driving dataset containing high-resolution vehicle trajectory data. To efficiently explore the hyperparameter space, we leverage Optuna[18], a state-of-the-art hyperparameter optimization framework, running multiple trials across different random seeds to ensure robust evaluation. Using functional ANOVA [19], we quantify both main and interaction effects of hyperparameters.

We find that in complex driving tasks, interaction effects dominate performance variation, significantly outweighing individual (univariate) influences, challenging assumptions from prior studies in simpler RL benchmarks [7], [19]. Furthermore, we identify the most influential hyperparameters and investigate the impact of random seed variation on performance and sensitivity outcomes.By focusing on a real-world autonomous driving benchmark, our study reveals key patterns in hyperparameter influenceand highlights the importance of developing robust, interaction-aware tuning strategies for high-stakes RL applications.

2. EXPERIMENTAL SETUP

2.1. Environment and RL Setup

We conduct our experiments using CommonRoad-RL[16], an open-source benchmark for autonomous driving built on top of Gym and Stable-Baselines. CommonRoad data converter, a tool included in the CommonRoad framework,transforms raw data from real-world traffic datasets into standardized CommonRoad scenarios, each defined by an ego vehicle, surrounding traffic, and a motion planning task with a specified goal region. We use the highD dataset[17].It provides 16.5 hours of highway vehicle trajectories recorded at 25 Hz (Δt = 0.04 s). Each trajectory is transformed into a 40-second CommonRoad scenario. The observation space includes information about the ego vehicle's state, nearby dynamic obstacles, road network geometry, nearby traffic participants, and task-specific indicators and goal region.

We employ continuous action spaces,enabling direct control of acceleration and steering. The ego vehicle's dynamics are defined by the point-mass vehicle model from CommonRoad-RL, which offers a computationally efficient approximation of motion while preserving essential kinematic properties for high-level planning.We adopt the CommonRoad-RL sparse, termination-based reward. At each step, a single event—goal reached, collision, off-road, or timeout—may occur; otherwise, the reward is zero(We disable the optional safe-distance term from CommonRoad-RL).Let $1_{goal,t}, 1_{collision,t}, 1_{offroad,t}, 1_{timeout,t} \in \{0\ , 1\}$ denote binary indicators that are 1 at time$t$if the corresponding termination event occurs and 0 otherwise (events are mutually exclusive). With coefficients $c_{goal}, c_{collision}, c_{offroad}, c_{timeout}$ given in Table 1,the per-step reward is

$$r_t = c_{goal} \cdot 1_{goal,t} + c_{collision} \cdot 1_{collision,t} + c_{offroad} \cdot 1_{offroad,t} + c_{timeout} \cdot 1_{timeout,t} \quad (1)$$

Table 1. Reward configurations

Reward Term	Coefficient
Goal Reached (c_{goal})	50.0
Collision($c_{collision}$)	-50.0
Off-Road($c_{offroad}$)	-20.0
Time-Out($c_{timeout}$)	-10.0

For training the Reinforcement Learning agent, we employ two of the most widely used RL algorithms in autonomous driving: Proximal Policy Optimization (PPO)[14] and Soft Actor-Critic (SAC) [15].PPO is an on-policy actor-critic method known for its stability and sample efficiency, achieved through a clipped surrogate objective that constrains policy updates and prevents destructive shifts during training. SAC, in contrast, is an off-policy algorithm that combines entropy maximization with actor-critic learning to encourage robust exploration and improve sample efficiency. We utilize the implementations of both algorithms provided by the Stable-Baselines library, which is built upon OpenAI Baselines and integrates with the CommonRoad-RL environment.

2.2. Hyperparameters Optimization Methodology

To study the effect of different hyperparameter configurations on reinforcement learning performance, we employ Optuna[18], a flexible and efficient hyperparameter optimization framework. We use random search as the sampling strategydisable pruning to obtain complete trials for fair comparison.

For each algorithm, we run 500 complete Optuna trials, using five different training seeds(100 trials per seed). Each trial is trained for 100,000 timesteps with no early stopping or pruning.Every 10,000 steps, the callback pauses learning and evaluates the current policy on a separate evaluation environment for five full episodes.For each episode, we compute the episodic return using the per-step sparse reward defined inSection 2.1.At each checkpoint, we average the five episodic returns to obtain the mean evaluation reward, and we keep the running maximum across all checkpoints within the trial (the best mean evaluation reward).The Optuna objective returns a cost defined as $cost = -best\_mean\_evaluation\_reward$ (direction: minimize); for reporting, we convert back to$best_mean_evaluation_reward = -trial.value$.

Throughout all trials, we maintain a consistent environment setup, including the vehicle model (point-mass), reward design, observation and action space, and we ensure that evaluation episodes are distinct from training rollouts (separate environment instance).Wetune a subset of PPO and SAC hyperparameters that are widely recognized to influence performance.The hyperparameter search spaces for PPO and SAC are provided in Table 2 (PPO) and Table 3 (SAC).

Table 2. PPO Hyperparameters Search Space

Hyperparameter	**Search Space**	**Type**
batch_size	[8, 16, 32, 64, 128, 256, 512]	Categorical
n_steps	[8, 16, 32, 64, 128, 256, 512, 1024, 2048]	Categorical
gamma	[0.9, 0.95, 0.98, 0.99, 0.995, 0.999, 0.9999]	Categorical
learning_rate	[1e-5, 1]	LogUniform Float
ent_coef	[1e-8, 0.1]	LogUniform Float
cliprange	[0.1, 0.2, 0.3, 0.4]	Categorical
noptepochs	[1, 5, 10, 20, 30, 50]	Categorical
lam (λ)	[0.8, 0.9, 0.92, 0.95, 0.98, 0.99, 1.0]	Categorical
max_grad_norm	[0.0, 1.0]	Uniform Float
vf_coef	[0.0, 1.0]	Uniform Float
cliprange_vf	[0.0, 0.5]	Uniform Float

Table 3. SAC Hyperparameters Search Space

Hyperparameter	**Search Space**	**Type**
gamma	[0.9, 0.95, 0.98, 0.99, 0.995, 0.999, 0.9999]	Categorical
learning_rate (lr)	[1e-5, 1]	Log-Uniform Float
tau	[1e-4,1.0]	Log-UniformFloat
batch_size	[16, 32, 64, 128, 256, 512]	Categorical
buffer_size	[1e4, 1e5, 1e6]	Categorical
learning_starts	[0, 10000]	Uniform Integer
train_freq	[1, 1000]	Uniform Integer
gradient_steps	[1, 10, 100, 300]	Categorical

3. RESULTS AND ANALYSIS

3.1. Hyperparameters Sensitivity Across Seeds

To evaluate the impact of hyperparameter choices and random seeds, we first examine the distribution of the best mean evaluation rewards across five random seeds, using 100 Optuna trials per seed (500 trials for each algorithm), as shown in Figure 1.

For PPO, distributions are tightly clustered and centred near zero. Median rewards range from approximately −0.7 to +3.3, with upper quartiles between +4.5 and +7.2 and maximum values consistently around +10. Although some configurations perform below zero (minimum rewards around −7.7 to −9.2), the overall spread is limited, and Interquartile ranges are compact (6.2–9.6 points) and overlap heavily across seeds.This indicates that PPO exhibits a relatively stable behaviour across hyperparameter configurations and weak seed sensitivity.

In contrast, SAC displays a much wider reward range. Median rewards span from −20 to −9 across seeds, with lower quartiles consistently near −26 and upper quartiles ranging from −6 up to +12. While SAC can achieve very high performance (maximum rewards up to +50), it also produces frequent failures, with minimum rewards often at −50. The wide interquartile range (20.5–38points) indicates a strong dependence on hyperparameter choices and moderate seed sensitivity; strong policies do occur, but they are relatively rare.

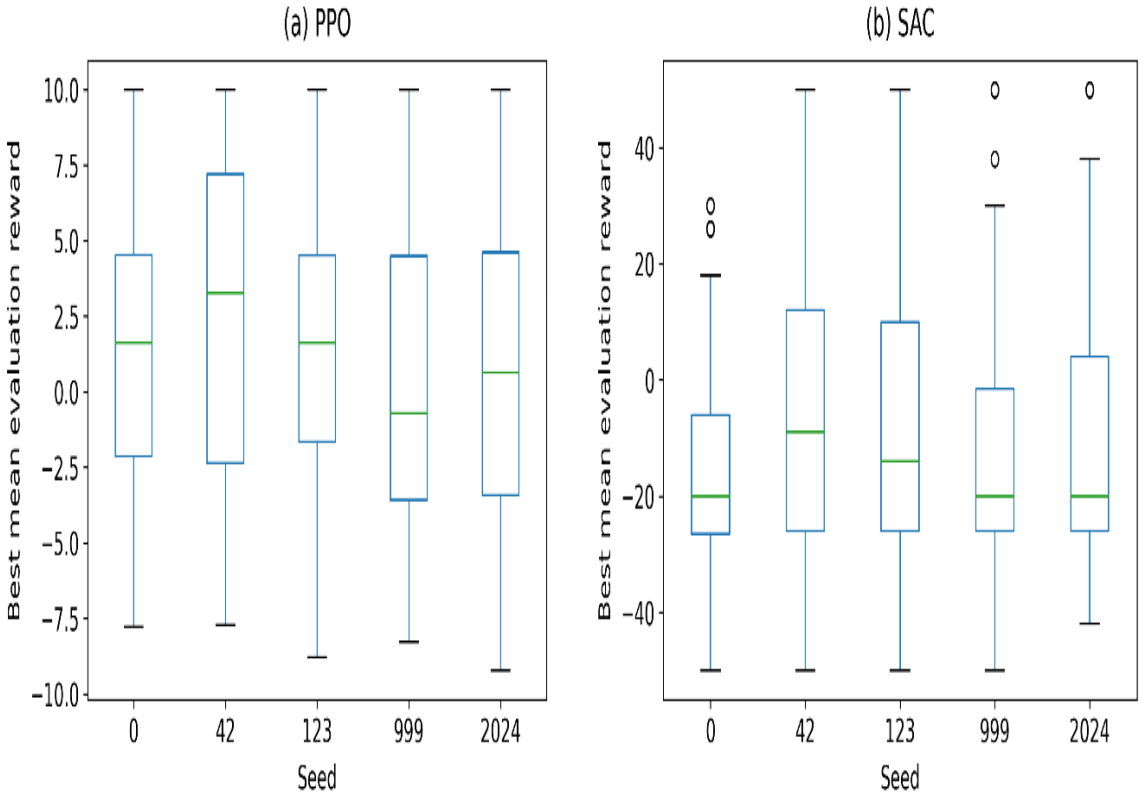

Figure 1.Best mean evaluation reward by random seed for (a) PPO and (b) SAC. For each algorithm we ran 5 seeds × 100 Optuna trials per seed (total 500 trials). At each seed, the box shows the interquartile range across trials, the horizontal line is the median, whiskers extend to 1.5×IQR, and points denote outliers.

These findings suggest that in highway autonomous driving scenarios, PPO could be more robust to hyperparameter choices, whereas SAC's performance varies more with seed and hyperparameters. In both cases, the within-seed spread dominates the between-seed shift, motivating the fANOVA analysis in the next section.

3.2. FANOVA Results

To assess the importance of each hyperparameter, we apply functional ANOVA[19]to the 500 completedoptimization trials for each algorithm. FANOVA maps hyperparameter configurations to performance using a random forest, then decomposes the total variance into main effects and higher-order interactions. To produce reliable importance estimates, we mitigate the randomness in the surrogate model (random forest) by repeating the FANOVA analysis using multiple random seeds (0, 42, 123, 999) and averaging the resulting importance values across runs. For both algorithms,we present the main effects in Figure 2 and the interaction effects in Figure 3. These plots illustrate how performance variance is attributed to individual hyperparameters and their combinations.

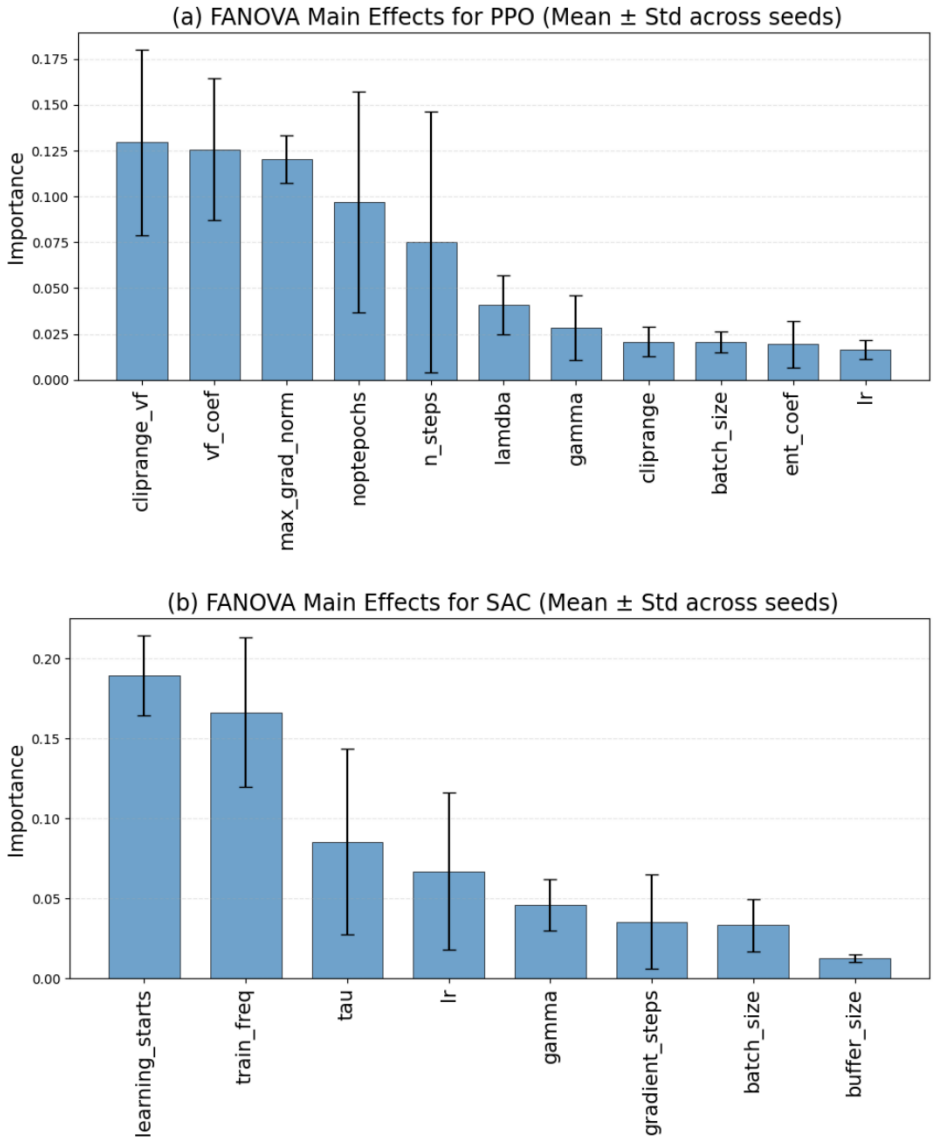

Figure 2. FANOVA main effects (mean ± standard deviation across five seeds) for (a) PPO and (b) SAC. Each bar shows the proportion of performance variance explained by a single hyperparameter.

Our analysis of Main and Interaction effects reveals distinct patterns of hyperparameter sensitivity for PPO and SAC, reflecting differences in their learning dynamics. PPO is most sensitive to hyperparameters related to value function learning and gradient stability, namely *cliprange_vf,vf_coef*, and *max_grad_norm,* emphasizing the importance of stable critic updates and controlled policy improvement. Moderate influence is observed for *noptepochs* and *n_steps*, while other parameters have relatively minor effects, indicating robustness to these settings within typical ranges.In contrast, SAC's performance is predominantly influenced by hyperparameters controlling experience usage and network update timing, specifically *learning_starts* and *train_freq*. The soft update rate *tau* and learning rate *lr* have moderate effects, whereas other hyperparameters exhibit minimal influence, highlighting SAC's relative robustness to network and buffer configurations. Overall, while PPO is highly sensitive to factors affecting stability and value learning, SAC is more sensitive to when and how experience is applied during training, reflecting the differing mechanisms underlying policy optimization in the two algorithms.

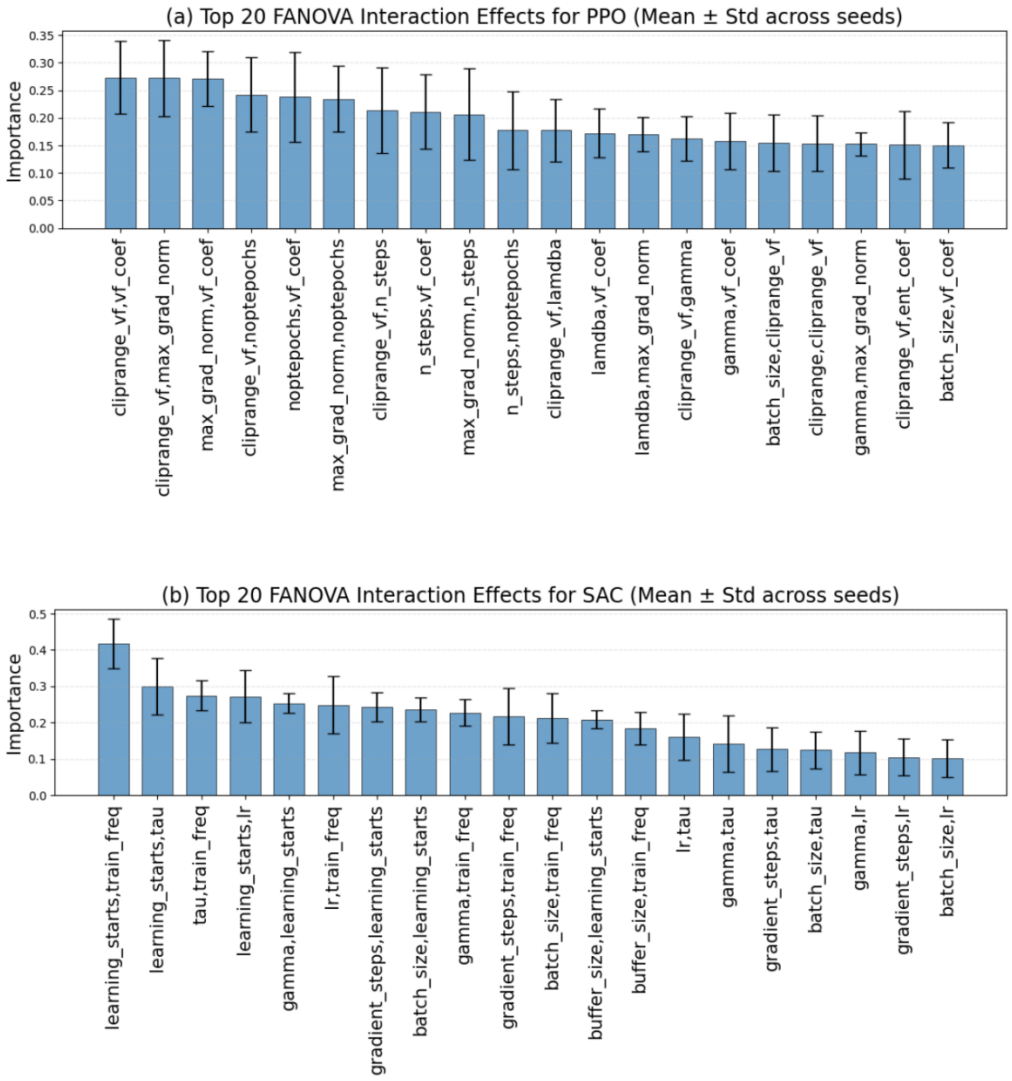

Figure 3. Top 20 FANOVA interaction effects (mean ± standard deviation over five seeds) for (a) PPO and (b) SAC. Bars indicate the magnitude of each interaction's contribution to performance variability.

A key finding of this analysis is that interaction effects overwhelmingly dominate the explained variance for both PPO and SAC. For PPO, the main effects account for only 0.0880 ± 0.0003 of variance, while interaction effects contribute 0.9120 ± 0.0003. Similarly, for SAC, main effects explain 0.117939 ± 0.000743 ofvariance, with interaction effects accounting for 0.882061 ± 0.000743. These results indicate that performance variability is primarily governed by complex interdependencies among hyperparameters, rather than isolated effects of individual parameters.To further illustrate the dominance of interaction effects, we present selected pairwise (2D) marginal plots produced by FANOVA for PPO in Figure 4 and for SAC in Figure 5.These plots depict how combinations of two hyperparameters jointly influence performance, revealing complex patterns that cannot be explained by main effects alone.

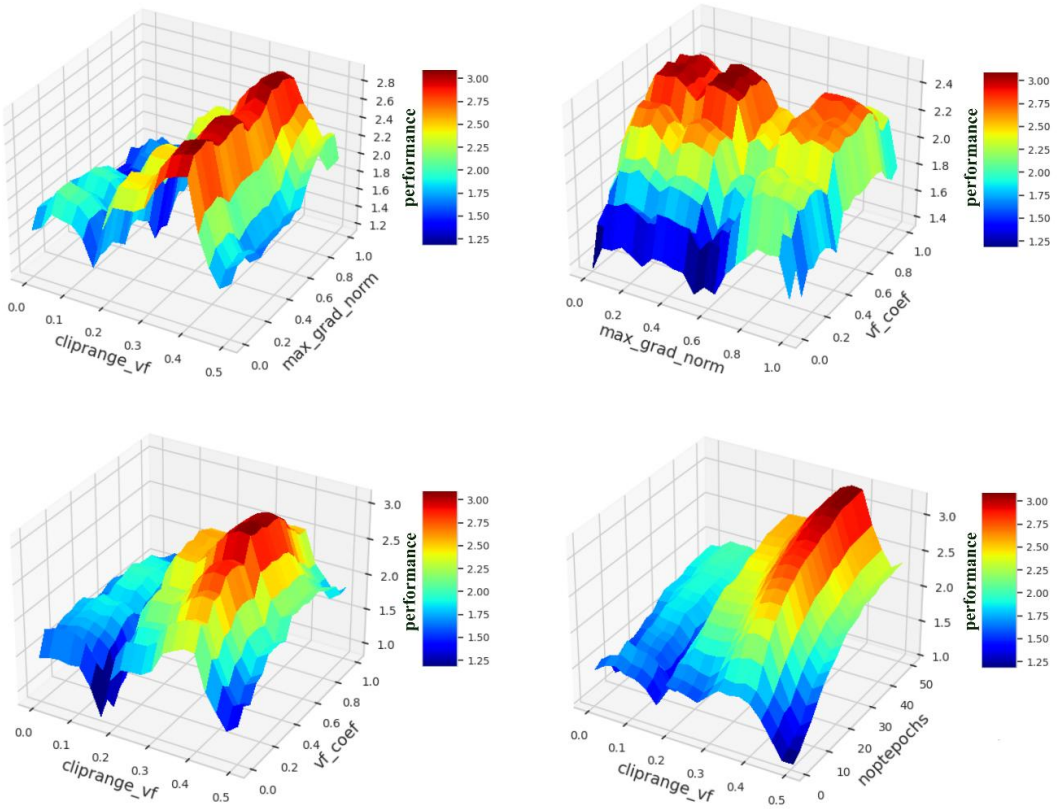

Figure 3.FANOVA pairwise marginal plots for PPO illustrating the four most influential hyperparameter interactions. The colour gradients depict the joint impact of each pair on expected reward. The pronounced non-linear patterns across plots highlight the strong and dominant role of interaction effects.

This finding contrasts with prior work conducted in other reinforcement learning environments. In [19], the authors observed that main effects explained between 20% and 88% of performance variance, while pairwise interactions reached at most 45%. Likewise, the study in[7] reported that one or two hyperparameters typically dominate, with little evidence of complex interaction patterns.This difference is likely due to the complex, high-dimensional nature of autonomous driving scenarios examined in our study which involves richer sensory inputs, and a greater need for coordination between learning components — all of which can amplify hyperparameter interdependencies and result in dominant interaction effects.

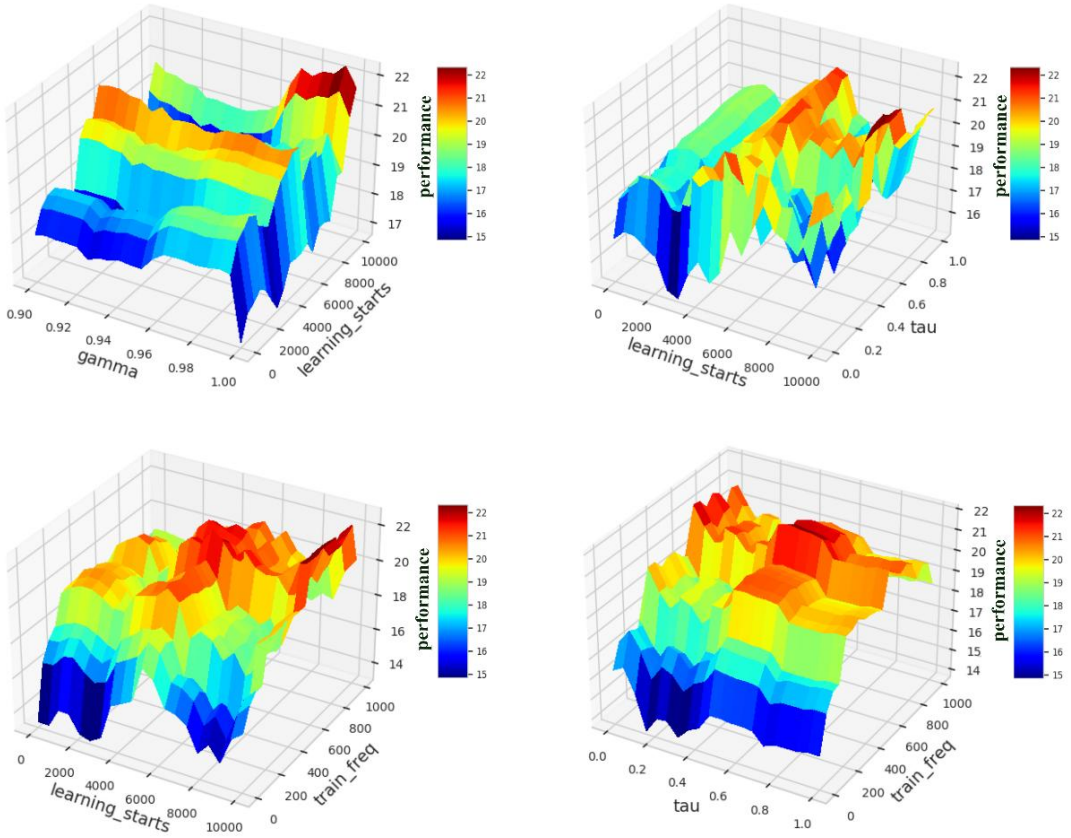

Figure 4. FANOVA pairwise marginal plots for SAC illustrating the four most influential hyperparameter interactions. The colour gradients depict the joint impact of each pair on expected reward The pronounced non-linear patterns across plots highlight the strong and dominant role of interaction effects.

3.3. Practical Implications for AutoRL

The dominance of interaction effects implies that tuning actor–critic RL algorithms such as PPO and SAC in autonomous-driving settings should go beyond univariate analyses and prioritize joint rather than one-at-a-time adjustments.In practice, for hyperparameter optimization this means: (i) jointly searching over the hyperparameter pairs/groups highlighted by FANOVA; (ii) using model-based search that can learn effects of hyperparameter combinations—for example, tree-based TPE/Random-Forest or Bayesian optimization with pairwise kernels—to propose new trials; (iii) optimizing a robust objective, e.g., the median return across multiple seeds and held-out scenarios (and reporting spread such as IQR), rather than a single best run; and (iv) turning the FANOVA interaction maps into lightweight joint-range guidelines (simple “keep-in” boxes covering the broad high-performance regions and excluding clearly poor/unstable areas) to guide future AutoRL pipelines toward well-behaved parts of the search space.(and, since interaction-heavy settings can learn slowly, apply pruning/early stopping only after consistent improvements across checkpoints and seeds).

This study is conducted in the CommonRoad-RL benchmark using a simplified point-mass vehicle model and the highD dataset, which—while realistic—primarily captures structured highway traffic. This setup offers control and reproducibility but does not fully reflect the complexity of urban or mixed-traffic conditions. Although we evaluate 500 Optuna trials across

five seeds per algorithm, the hyperparameter space for PPO and SAC is high-dimensional, so we focus on a widely recognized subset of influential parameters for tractability. For SAC specifically, *ent_coef* and *target_entropy* are excluded from FANOVA because they introduce mixed/conditional variable types (e.g., automatic vs. fixed temperature), complicating variance attribution under our protocol. Fixed training budgets (100,000 steps per trial) may limit exploration of slower-converging configurations. Finally, our FANOVA decomposition emphasizes main and pairwise effects and may underrepresent higher-order dependencies among hyperparameters.

4. CONCLUSIONS

This study provides a comprehensive empirical evaluation of hyperparameter sensitivity for PPO and SAC in autonomous driving tasks using the CommonRoad-RL framework and the highD dataset. Our analysis demonstrates that performance variability in both algorithms is overwhelmingly dominated by hyperparameter interactions, accounting for over 91% of variance in PPO and 88% in SAC, while main effects contribute only marginally. This highlights that the impact of hyperparameters cannot be understood in isolation; rather, complex interdependencies between parameters drive learning outcomes in high-dimensional, safety-critical environments.

PPO performance is driven by value function learning and gradient stability (*cliprange_vf*, *vf_coef*, *max_grad_norm*), with moderate influence from hyperparameters such as*noptepochs* and *n_steps*. In contrast, SAC is most sensitive to experience usage and training schedule (*learning_starts*, *train_freq*), with moderate effects from hyperparameters such as*tau* and *lr*. This contrast reflects their differing optimization mechanisms, where PPO depends on stable value updates, while SAC relies on effective use of collected experience.

The significant dominance of interaction effects in both algorithms highlights the importance of moving beyond univariate analyses when studying hyperparameter sensitivity and optimization in real-world, safety-critical RL domains. Practically, this calls for interaction-aware tuning: (i) joint search over the pairs/groups surfaced by FANOVA, (ii) model-based HPO that can learn effects of hyperparameter combinations (e.g., TPE/Random-Forest or BO with pairwise kernels), (iii) robust objectives (median across multiple seeds and held-out scenarios, with spread such as IQR), and (iv) lightweight joint-range guidelines derived from FANOVA interaction maps to steer AutoRL away from brittle regions. Future work will extend the analysis beyond highways to richer, more complex settings (e.g., urban driving and intersections), compare on-policy and off-policy methods, broaden the algorithmic scope (e.g., TRPO, A2C, TD3/DDPG, DQN, and model-based RL), and progress from simulation benchmarks to real-world evaluations, while integrating interaction-aware tuning with safety constraints and online adaptation.

REFERENCES

[1] B. R. Kiran *et al.*, “Deep Reinforcement Learning for Autonomous Driving: A Survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, June 2022, doi: 10.1109/TITS.2021.3054625.

[2] A. Irshayyid, J. Chen, and G. Xiong, “A review on reinforcement learning-based highway autonomous vehicle control,” *Green Energy Intell. Transp.*, vol. 3, no. 4, p. 100156, Aug. 2024, doi: 10.1016/j.geits.2024.100156.

[3] Y. Chen, C. Ji, Y. Cai, T. Yan, and B. Su, “Deep Reinforcement Learning in Autonomous Car Path Planning and Control: A Survey,” Mar. 30, 2024, *arXiv*: arXiv:2404.00340. doi: 10.48550/arXiv.2404.00340.

[4] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, “Deep Reinforcement Learning that Matters,” Jan. 30, 2019, *arXiv*: arXiv:1709.06560. doi: 10.48550/arXiv.1709.06560.

[5] M. Andrychowicz *et al.*, "What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study," June 10, 2020, *arXiv*: arXiv:2006.05990. doi: 10.48550/arXiv.2006.05990.

[6] L. Engstrom *et al.*, "Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO," May 25, 2020, *arXiv*: arXiv:2005.12729. doi: 10.48550/arXiv.2005.12729.

[7] T. Eimer, M. Lindauer, and R. Raileanu, "Hyperparameters in Reinforcement Learning and How To Tune Them," June 02, 2023, *arXiv*: arXiv:2306.01324. doi: 10.48550/arXiv.2306.01324.

[8] J. Adkins, M. Bowling, and A. White, "A Method for Evaluating Hyperparameter Sensitivity in Reinforcement Learning," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 124820–124842, Dec. 2024.

[9] A. Mohan, C. Benjamins, K. Wienecke, A. Dockhorn, and M. Lindauer, "AutoRL Hyperparameter Landscapes," June 05, 2023, *arXiv*: arXiv:2304.02396. doi: 10.48550/arXiv.2304.02396.

[10] J. Parker-Holder *et al.*, "Automated Reinforcement Learning (AutoRL): A Survey and Open Problems," *J. Artif. Intell. Res.*, vol. 74, pp. 517–568, June 2022, doi: 10.1613/jair.1.13596.

[11] J. K. H. Franke, G. Köhler, A. Biedenkapp, and F. Hutter, "Sample-Efficient Automated Deep Reinforcement Learning," Mar. 17, 2021, *arXiv*: arXiv:2009.01555. doi: 10.48550/arXiv.2009.01555.

[12] G. Shala, S. P. Arango, A. Biedenkapp, F. Hutter, and J. Grabocka, "AutoRL-Bench 1.0," 2022. Accessed: June 15, 2025. [Online]. Available: https://www.semanticscholar.org/paper/AutoRL-Bench-1.0-Shala-Arango/dd9e6908ce14f99203fc3f13edcdb4f7236a1153

[13] M. Kiran and M. Ozyildirim, "Hyperparameter Tuning for Deep Reinforcement Learning Applications," Jan. 26, 2022, *arXiv*: arXiv:2201.11182. doi: 10.48550/arXiv.2201.11182.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 28, 2017, *arXiv*: arXiv:1707.06347. doi: 10.48550/arXiv.1707.06347.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Aug. 08, 2018, *arXiv*: arXiv:1801.01290. doi: 10.48550/arXiv.1801.01290.

[16] X. Wang, H. Krasowski, and M. Althoff, "CommonRoad-RL: A Configurable Reinforcement Learning Environment for Motion Planning of Autonomous Vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sept. 2021, pp. 466–472. doi: 10.1109/ITSC48978.2021.9564898.

[17] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 2118–2125. doi: 10.1109/ITSC.2018.8569552.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.

[19] F. Hutter, H. Hoos, and K. Leyton-Brown, "An Efficient Approach for Assessing Hyperparameter Importance," in *Proceedings of the 31st International Conference on Machine Learning*, PMLR, Jan. 2014, pp. 754–762. Accessed: June 15, 2025. [Online]. Available: https://proceedings.mlr.press/v32/hutter14.html