PER-Falcon: Positive-Episode Replay for Future-Aware Social Navigation

Zihao Zhang[1], Yu Zhong[1,2], Enzhu Gao[3], Xinhan Zheng[4], Xueting Wang[4], Shouming Li[5], Yunkai Gao[3], Siming Lan[3], Mingfei Han[6], Xing Hu[1,2]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3]Institute of AI for Industries
[4]University of Science and Technology of China [5]Beijing University of Technology
[6]Mohamed Bin Zayed University of Artificial Intelligence

{zhangzihao, zhongyu21s, huxing}@ict.ac.cn,

Abstract

*Navigating in crowded public spaces is a challenging task in robotics. Previous work, such as Falcon, has conducted in-depth exploration of this problem. Our approach, PER-Falcon, follows the Falcon framework and introduces a **P**ositive **E**pisode **R**eplay strategy to prioritize high-value episodes during the optimization process in reinforcement learning. Specifically, valuable navigating trajectories are periodically identified, collected into a secondary pool, and reused through a PPO update. Our approach PER-Falcon ranks 1st place in the challenge with remarkable 7 points improvements compared with the official baseline on the phase-2 test set.*

1. Introduction

In recent years, the field of embodied navigation is on the cusp of research due to its potential significance for service robotics [1–10]. One main challenge is the application to crowded public spaces, where human-robot interaction norms and collision avoidance are crucial. Therefore, socially compliant navigation is a cornerstone for service robots that are expected to coexist with humans in photo-realistic indoor environments [11, 12].

One pivotal advancement in this area is the proposal of Falcon that explicitly achieves social compliance by encouraging socially comfortable behaviors via trajectory-prediction auxiliary tasks and obstruction penalties [2]. This approach contributes a structured and effective framework for navigating photo-realistic scenes shared with humans. However, Falcon adopts uniform trajectory-sampling during training and fails to prioritize those valuable episodes that contain more efficient maneuvers and informative context. This limitation impedes learning of skilled and complex behaviors (such as pedestrian yieldings) required for the social navigation task, potentially leading to suboptimal performance.

To address this issue in the training strategy, we propose **PER-Falcon**, employing the data-efficient strategy Positive-Episode Replay (PER), which inherently focuses on episodes with high returns. We build our method on top of the framework of Falcon and optimize the training process by periodically storing high-return episodes whose undiscounted return exceeds a pre-defined threshold and reusing them for one additional auxiliary PPO epoch. Specifically, we design two key modules, the Secondary Batch Construction module and the Auxiliary PPO Update module, to achieve this.

Our approach ranked in the first place in the 2025 RoboSense Challenge Track 2 [13], outperforming the official baseline method with 7 points of total score. The experimental results demonstrate effectiveness and foresight in real-world applications of our approach, marking significant progress in the field of social navigation.

2. Methodology

Our solution is built upon Falcon, an RL-based policy strategy enabling agents to avoid humans while navigating to the destination. We introduce the main pipeline of our method in Fig 1. Please also refer to the original Falcon [14] paper for more details. To better exploit high-value episodes, we propose two key modules: the Secondary Batch Construction module and the Auxiliary PPO Epoch module to realize a PER strategy.

2.1. Overall Pipeline

As shown in Fig 1 (a), we solve the social navigation with a PPO-based network performing in a recurrent manner. Specifically, for each timestep t, we perform PPO optimiza-

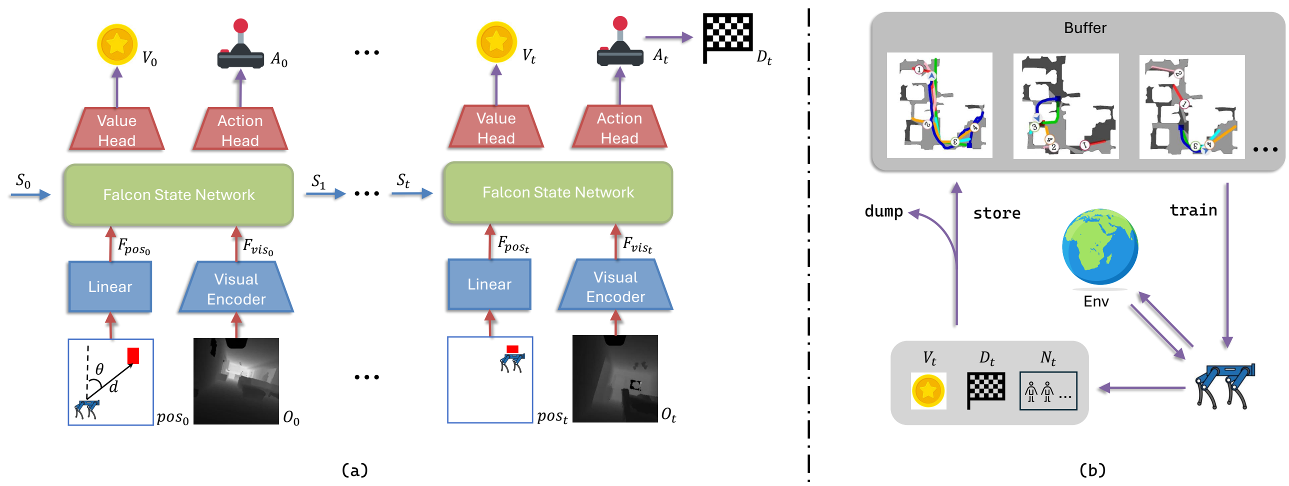

Figure 1. Overall pipeline of the proposed PER-Falcon. (a). Our approach adopts a recurrent manner. We take the GPS information and the observed depth maps as inputs and produce the action at each timestep. (b). The main scheme of PER. We store those valuable experiences and dump others after completing a trajectory. The stored trajectories will be adopted for an additional training epoch.

tion over the most recent on-policy batch:

$$\mathcal{B} = \{(s_t, a_t, r_t, s_{t+1}, \log \pi_{\theta_{\text{old}}}, m_t)\}_{t=1}^{T},$$

where s_t and a_t denote state and action at timestep t, r_t denotes the immediate reward, s_{t+1} denotes the next state, $\log \pi_{\theta_{\text{old}}}$ denotes the log-probability under the old policy, and m_t denotes the continuation mask used by the RNN backbone; T denotes the rollout horizon. After the update, $\mathcal{B}$ is discarded, and a new batch is collected. While clipping approximately guarantees monotonic improvement, transitions that once yielded a high advantage are lost even if they remain informative for the new policy.

2.2. Secondary Batch Construction

The goal of secondary batch construction is to reuse high-value episodes for an additional policy update. Specifically, let an episode i be the sequence

$$\tau_i = \{(s_t, a_t, r_t, s_{t+1}, m_t)\}_{t=0}^{T_i-1}, \quad R_i = \textstyle\sum_{t=0}^{T_i-1} r_t,$$

where s_t (state), a_t (action), r_t (immediate reward), s_{t+1} (next state), and $m_t \in \{0,1\}$ (RNN continuation mask, 1=continue) are collected at timestep t, and R_i is the undiscounted return. After every rollout, we build the positive subset

$$\mathcal{P} = \big\{i \in [0, N-1] \;\big|\; \text{episode } i \text{ is terminal } \textbf{and } R_i \geq \rho\big\},$$

where N denotes the number of parallel environments (batch size), ρ denotes the scalar return threshold. In our experiments, we set $\rho = 10$, corresponding to approximately 75 percentile of the return distribution observed during the initial training phase. This value was fixed without any scheduling. For every $i \in \mathcal{P}$, we extract the corresponding trajectory slice from the rollout buffer and obtain $\tilde{\tau}_i$ via frame-by-frame deep copy.

$$\tilde{m}_t \leftarrow 1, \quad \forall t,$$

We set the continuation mask $\tilde{m}_t$ to 1 for all time steps t. This setting prevents the RNN hidden state from resetting, thereby preserving the structural integrity of the replay buffer without altering the batch's mathematical content. This approach is an implementation convenience that avoids accidental hidden-state clearance and logically treats the entire trajectory as an intermediate segment of one long episode.

All other variables including the old log-probability $\log \pi_{\theta_{\text{old}}}$ are preserved, so the probability ratio $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ is computed with the cached logits and remains within the PPO clipping range No additional importance-sampling weight is required. In multi-agent settings the procedure is repeated independently per agent.

2.3. Auxiliary PPO Epoch

The auxiliary PPO epoch aims to update the original PPO using prioritized transitions. To achieve this, selected transitions are cloned and concatenated into a secondary batch $\mathcal{B}_{\text{pos}} = \bigcup_{i\in\mathcal{P}} \tilde{\tau}_i$. The auxiliary PPO update applies gradients *sequentially* after the primary epoch executed on $\mathcal{B}_{\text{pos}}$. *Critically*, this auxiliary update not only preserves network parameters across updates, but also maintains stochasticity via random mini-batch shuffling. Resulting losses are logged for analysis but do not affect core training dynamics.

2.4. Algorithm Summary

Algorithm 1 provides pseudocode for integrating PER into standard PPO training. The key additions are:

- **Conditional execution**: Auxiliary updates occur every K parameter update ($K = 20$ in our experiments)

- **Mask overwriting**: All masks in replayed episodes are set to 1
- **Multi-agent support**: Each agent processes its own positive episodes independently
- **Loss monitoring**: Auxiliary update losses are recorded for analysis

Algorithm 1 Positive-Episode Replay

Require: threshold ρ, auxiliary interval K

initialize policy π_θ, rollout buffer $\mathcal{B}$
for update step $n = 1, 2, \ldots$ **do**
  $\mathcal{B} \leftarrow$ COLLECTROLLOUT(π_θ) ▷ standard PPO data collection
  $\mathcal{P} \leftarrow \{i \mid \text{episode } i \text{ is terminal} \wedge R_i \geq \rho\}$
  $\mathcal{L}_{\text{primary}} \leftarrow$ PPOUPDATE($\mathcal{B}$) ▷ primary PPO update
  if $n \bmod K = 0$ **and** $\mathcal{P} \neq \emptyset$ **then**
    $\mathcal{B}_{\text{pos}} \leftarrow \emptyset$
    for each agent a in multi-agent system **do**
      Extract trajectories $\{\tau_i^a\}_{i \in \mathcal{P}}$ from agent a's rollout buffer
      for each trajectory τ_i^a **do**
        $\tilde{\tau}_i^a \leftarrow$ DEEPCOPY(τ_i^a) ▷ frame-wise `.clone()`
        Set $\tilde{m}_t = 1$ for all transitions in $\tilde{\tau}_i^a$ ▷ buffer format only
      end for
      $\mathcal{B}_{\text{pos}}^a \leftarrow \bigcup_{i \in \mathcal{P}} \tilde{\tau}_i^a$
      $\mathcal{L}_{\text{aux}}^a \leftarrow$ PPOUPDATE($\mathcal{B}_{\text{pos}}^a$) ▷ auxiliary update per agent
    end for
    LOGLOSSES($\mathcal{L}_{\text{aux}}$) ▷ record for monitoring
  end if
end for

3. Experiments

3.1. Dataset

We use the official data provided by the *RoboSense Challenge 2025* [15] held at IROS 2025. This competition builds upon the legacy of the *RoboDepth Challenge 2023* [16, 17] at ICRA 2023 and the *RoboDrive Challenge 2024* [18, 19] at ICRA 2024, continuing the collective effort to advance robust and scalable robot perception. Each track in this competition is grounded on an established benchmark designed for evaluating real-world robustness and generalization [20–25].

The Social-HM3D dataset contains 844 high-fidelity 3D indoor scenes reconstructed from real building-scale scans; together with Social-MP3D (72 scenes), they constitute the largest benchmark suite for social navigation to date. By adopting an adaptive-density mechanism, Social-HM3D generates realistically distributed virtual humans and significantly alleviates domain bias caused by over-crowded or under-crowded scenarios, providing higher physical and social plausibility for algorithm evaluation.

3.2. Experimental Setup

We follow the open-source work of Gong et al. [21] and conduct experiments. Our model is trained on the Social-HM3D training split and evaluated on the public Phase-1 test set as well as the hidden Phase-2 test set released by the RoboSense Challenge – SocialNav Track, ensuring full reproducibility and compliance with the official protocol.

Evaluation Criteria. We adopt the unified metrics published by the organisers. Task completion and social compliance are jointly evaluated through the following indicators:

- **Success Rate (SR):** Percentage of episodes in which the robot stops within 1.0 m of the goal and the goal is oracle-visible (i.e. visible from the final pose without further movement).
- **SPL (Success weighted by Path Length):** Path efficiency w.r.t. the shortest path,

$$\text{SPL} = \frac{1}{N} \sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)},$$

where l_i is the shortest-path distance, p_i the actual path length, and $S_i = 1$ if successful.
- **PSC (Personal-Space Compliance):** Ratio of timesteps in which the robot keeps at least 0.5 m away from every human, reflecting socially appropriate behaviour.
- **H-Coll (Human-Collision Rate):** Percentage of episodes that contain any human collision; such episodes are counted as failures and degrade SR and PSC.

The overall score is computed by

$$\text{Total} = 0.4 \cdot \text{SR} + 0.3 \cdot \text{SPL} + 0.3 \cdot \text{PSC}.$$

3.3. Implementation Details

We strictly follow the hyper-parameters released in the official Falcon repository and make only one change to the reward function: the collide-human penalty is increased from -0.015 to -0.017 to further discourage collisions with humans.

For the Positive-Episode Replay (PER) module we retain Falcon's 128-step rollout setting and trigger PER every 20 parameter updates. The return threshold is set to 10, identical to the success bonus awarded by Falcon upon task completion, ensuring a clear separation between positive and negative trajectories (see §3.5 for justification).

Our PER-Falcon model is initialised from the public Falcon checkpoint and undergoes 2 000 fine-tuning updates without any extra data or manual annotation. Training is performed with Habitat-sim's built-in DD-PPO on 4 NVIDIA V100 GPUs; total training time was approximately 40 hours.

Table 1. Performance comparison on the Phase-2 test set

Method	SR ↑	SPL ↑	PSC ↑	H-Coll ↓	Total ↑
Baseline	0.5400	0.4997	0.8630	0.3920	0.6248
Ours	**0.6600**	**0.5977**	**0.8629**	**0.3240**	**0.7022**

Table 2. Ablations on return thresholds for experience replay in Social-HM3D

Method	SR ↑	SPL ↑	PSC ↑	H-Coll ↓	Total ↑
Baseline[1]	0.5455	0.4986	0.8931	0.4269	0.6357
NER (return < -0.05)	0.5465	0.4984	0.8930	0.4315	0.6360
NER (return < 0)	0.5501	0.4914	0.8972	0.4177	0.6366
PER (return > 10)	**0.5943**	**0.5311**	**0.8993**	**0.3910**	**0.6668**
$R_{0.015}$(Baseline)	0.5455	0.4986	0.8931	0.4269	0.6357
$R_{0.030}$	0.5695	0.5050	0.8966	0.4075	0.6483
$R_{0.020}$	0.5685	0.5112	0.8955	0.4177	0.6494
$R_{0.017}$	**0.5952**	**0.5319**	**0.8994**	**0.3845**	**0.6675**
w/o Human	0.9291	0.8519	1.0000	0.0000	0.9272

[1] Local evaluation of the official Falcon checkpoint to ensure fair comparison.

3.4. Comparison with Baseline Method

Table 1 reports Phase-2 results. PER-Falcon outperforms the baseline on all metrics. The PER-Falcon shows a 12 percentage improvements on the success rate and 0.0774 improvements over the total score.

3.5. Ablation Study

To verify the effectiveness of Positive-Episode Replay (PER) and to justify replaying only *positive* trajectories, we conduct a systematic ablation on Social-HM3D while keeping all other settings identical to §3.3. Results are shown in Table 2.

Analysing the episode-return distribution of Falcon reveals a clear bimodal pattern: a major peak at $\mu = 0$ (failed episodes) and a secondary peak at $\mu = 10$ (successful episodes). Consequently, we replay only trajectories with return > 10 to exploit high-quality success sequences. To rigorously evaluate the importance of selecting *positive* episodes, we include two Negative-Episode Replay (NER) baselines that replay trajectories *below* specified return thresholds.

The results provide a clear justification for our approach. As shown in Table 2, both NER variants achieve only marginal improvements over the baseline, with Success Rates (SR) of 0.5465 and 0.5501, virtually indistinguishable from the baseline (0.5455). In contrast, PER (return> 10) delivers substantially superior performance, boosting the SR to **0.5943** and SPL to **0.5311**, while simultaneously reducing human collisions (H-Coll) to **0.3910**. This stark performance gap demonstrates that replaying negative episodes provides negligible benefits, whereas selectively replaying high-return positive episodes is crucial for achieving significant performance gains.

To further optimize the agent's social navigation capability, we investigate the sensitivity of the human collision penalty weight, which we adjusted from the baseline value of -0.015 to -0.017 in our main experiments. This ablation study systematically evaluates different penalty weights while keeping all other experimental settings consistent. As shown in the lower part of Table 2, the choice of collision penalty weight has a significant impact on navigation performance. Our analysis reveals two key findings: First, the penalty weight exhibits a clear optimal range around -0.017. While both $R_{0.02}$ and $R_{0.03}$ show improved Success Rates (0.5685 and 0.5695 respectively) compared to the baseline $R_{0.015}$ (0.5400), the proposed $R_{0.017}$ achieves the best performance across all metrics, with a Success Rate of **0.5952** and the lowest human collision rate (**0.3845**). This suggests that moderate penalty values effectively discourage collisions without being overly conservative, thereby maintaining navigation efficiency.

Second, the comparison with the human-free environment (w/o Human) provides crucial insights. The nearly perfect performance (SR: 0.9291) in scenarios without humans demonstrates that human interactions constitute the primary challenge in social navigation. The significant performance gap between human-present and human-free conditions (approximately 33 percentage points in SR) underscores the critical importance of properly handling social interactions.

These results collectively indicate that most failure cases in social navigation are indeed caused by human collisions, making the collision penalty weight a crucial hyperparameter. The performance appears particularly sensitive around the optimal value, with satisfactory results observed only near -0.017. This empirical finding justifies our selection of -0.017 as the human collision penalty weight in all experiments.

4. Conclusion

In this work, we present PER-Falcon, an RL-based framework for social navigation built upon the Falcon baseline. To better utilize valuable experiences and promote the learning of complex social behaviors, we introduce the Positive Episode Replay strategy, which selectively replays high-return trajectories to optimize the training process. Our comprehensive experiments demonstrate the effectiveness of PER, showing significant performance improvements over both the baseline and negative replay strategies across all key metrics. The ablation study further reveals the sensitivity of performance to the human collision penalty weight, with systematic evaluation identifying -0.017 as the optimal value. Through these enhancements, our method achieved first place in the 2025 RoboSense Challenge with a total score of 0.7022 on the phase 2 test split, validating the practical efficacy of our approach for real-world social navigation tasks. Future work will explore adaptive replay strategies with dynamic threshold adjustment and investigate integrating PER with other advanced experience replay techniques.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[2] Zeying Gong et al. Stairway to success: An online floor-aware zero-shot object-goal navigation framework via LLM-driven coarse-to-fine exploration. *arXiv preprint arXiv:2505.23019*, 2025.

[3] Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. Prioritized semantic learning for zero-shot instance navigation. In *European Conference on Computer Vision*, pages 161–178. Springer, 2024.

[4] Runnan Chen et al. CLIP2Scene: Towards label-efficient 3D scene understanding by CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[5] Rong Li et al. SeeGround: See and ground for zero-shot open-vocabulary 3D visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3707–3717, 2025.

[6] Runnan Chen et al. Towards label-free scene understanding by vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 75896–75910, 2023.

[7] Xuzhi Wang, Xinran Wu, Song Wang, et al. Monocular semantic scene completion via masked recurrent networks. In *IEEE/CVF International Conference on Computer Vision*, pages 24811–24822, 2025.

[8] Xiang Xu et al. Beyond one shot, beyond one perspective: Cross-view and long-horizon distillation for better LiDAR representations. In *IEEE/CVF International Conference on Computer Vision*, pages 25506–25518, 2025.

[9] Xiaoshuai Hao, Mengchuan Wei, Yifan Yang, et al. Is your HD map constructor reliable under sensor corruptions? In *Advances in Neural Information Processing Systems*, volume 37, pages 22441–22482, 2024.

[10] Xiaoshuai Hao et al. SafeMap: Robust HD map construction from incomplete observations. In *International Conference on Machine Learning*, pages 22091–22102. PMLR, 2025.

[11] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[12] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 12(3):1–39, 2023.

[13] RoboSense Challenge 2025 Organizers. Robosense challenge 2025: Track 2 - social navigation. https://robosense2025.github.io/track2, 2025.

[14] Zeying Gong, Tianshuai Hu, Ronghe Qiu, and Junwei Liang. From cognition to precognition: A future-aware framework for social navigation, 2025.

[15] Lingdong Kong, Shaoyuan Xie, Zeying Gong, Ye Li, Meng Chu, Ao Liang, Yuhao Dong, Tianshuai Hu, Ronghe Qiu, Rong Li, Hanjiang Hu, Dongyue Lu, Wei Yin, Wenhao Ding, Linfeng Li, Hang Song, Wenwei Zhang, Yuexin Ma, Junwei Liang, Zhedong Zheng, Lai Xing Ng, Benoit R. Cottereau, Wei Tsang Ooi, Ziwei Liu, Zhanpeng Zhang, Weichao Qiu, Wei Zhang, Ji Ao, Jiangpeng Zheng, Siyu Wang, Guang Yang, Zihao Zhang, Yu Zhong, Enzhu Gao, Xinhan Zheng, Xueting Wang, Shouming Li, Yunkai Gao, Siming Lan, Mingfei Han, Xing Hu, Dusan Malic, Christian Fruhwirth-Reisinger, Alexander Prutsch, Wei Lin, Samuel Schulter, Horst Possegger, Linfeng Li, Jian Zhao, Zepeng Yang, Yuhang Song, Bojun Lin, Tianle Zhang, Yuchen Yuan, Chi Zhang, Xuelong Li, Youngseok Kim, Sihwan Hwang, Hyeonjun Jeong, Aodi Wu, Xubo Luo, Erjia Xiao, Lingfeng Zhang, Yingbo Tang, Hao Cheng, Renjing Xu, Wenbo Ding, Lei Zhou, Long Chen, Hangjun Ye, Xiaoshuai Hao, Shuangzhi Li, Junlong Shen, Xingyu Li, Hao Ruan, Jinliang Lin, Zhiming Luo, Yu Zang, Cheng Wang, Hanshi Wang, Xijie Gong, Yixiang Yang, Qianli Ma, Zhipeng Zhang, Wenxiang Shi, Jingmeng Zhou, Weijun Zeng, Kexin Xu, Yuchen Zhang, Haoxiang Fu, Ruibin Hu, Yanbiao Ma, Xiyan Feng, Wenbo Zhang, Lu Zhang, Yunzhi Zhuge, Huchuan Lu, You He, Seungjun Yu, Junsung Park, Youngsun Lim, Hyunjung Shim, Faduo Liang, Zihang Wang, Yiming Peng, Guanyu Zong, Xu Li, Binghao Wang, Hao Wei, Yongxin Ma, Yunke Shi, Shuaipeng Liu, Dong Kong, Yongchun Lin, Huitong Yang, Liang Lei, Haoang Li, Xinliang Zhang, Zhiyong Wang, Xiaofeng Wang, Yuxia Fu, Yadan Luo, Djamahl Etchegaray, Yang Li, Congfei Li, Yuxiang Sun, Wenkai Zhu, Wang Xu, Linru Li, Longjie Liao, Jun Yan, Benwu Wang, Xueliang Ren, Xiaoyu Yue, Jixian Zheng, Jinfeng Wu, Shurui Qin, Wei Cong, and Yao He. The RoboSense challenge: Sense anything, navigate anywhere, adapt across platforms. https://robosense2025.github.io, 2025.

[16] Lingdong Kong, Yaru Niu, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit Cottereau, Liangjun Zhang, Hesheng Wang, Wei Tsang Ooi, Ruijie Zhu, Ziyang Song, Li Liu, Tianzhu Zhang, Jun Yu, Mohan Jing, Pengwei Li, Xiaohua Qi, Cheng Jin, Yingfeng Chen, Jie Hou, Jie Zhang, Zhen Kan, Qiang Lin, Liang Peng, Minglei Li, Di Xu, Changpeng Yang, Yuanqi Yao, Gang Wu, Jian Kuai, Xianming Liu, Junjun Jiang, Jiamian Huang, Baojun Li, Jiale Chen, Shuang Zhang, Sun Ao, Zhenyu Li, Runze Chen, Haiyong Luo, Fang Zhao, and Jingze Yu. The RoboDepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023.

[17] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit R. Cottereau, and Wei Tsang Ooi. RoboDepth: Robust out-of-distribution depth estimation under corruptions. In *Advances in Neural Information Processing Systems*, volume 36, pages 21298–21342, 2023.

[18] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Yaru Niu, Wei Tsang Ooi, Benoit R. Cottereau, Lai Xing Ng, Yuexin Ma, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, Weichao Qiu, Wei Zhang, Xu Cao, Hao Lu, Ying-Cong Chen, et al. The RoboDrive challenge: Drive anytime anywhere in any condition. *arXiv preprint arXiv:2405.08816*, 2024.

[19] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3878–3894, 2025.

[20] Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are VLMs ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives. In *IEEE/CVF International Conference on Computer Vision*, pages 6585–6597, 2025.

[21] Zeying Gong, Tianshuai Hu, Ronghe Qiu, and Junwei Liang. From cognition to precognition: A future-aware framework for social navigation. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9122–9129, 2024.

[22] Ye Li, Lingdong Kong, Hanjiang Hu, Xiaohao Xu, and Xiaonan Huang. Is your LiDAR placement optimized for 3D scene understanding? In *Advances in Neural Information Processing Systems*, volume 37, pages 34980–35017, 2024.

[23] Meng Chu, Zhedong Zheng, Wei Ji, Tingyu Wang, and Tat-Seng Chua. Towards natural language-guided drones: GeoText-1652 benchmark with spatial relation matching. In *European Conference on Computer Vision*, pages 213–231, 2024.

[24] Ao Liang et al. Perspective-invariant 3D object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 27725–27738, 2025.

[25] Rong Li, Yuhao Dong, Tianshuai Hu, Ao Liang, et al. 3EED: Ground everything everywhere in 3D. *arXiv preprint arXiv:2511.01755*, 2025.