Proximal State Nudging: Reducing Skill Atrophy from AI Assistance

Megha Srivastava [1] **Jonathan Ouyang** [2] **Eric Zhou** [2] **Andrew Silva** [3] **Emily Sumner** [3] **Dorsa Sadigh** [1]
Yuchen Cui [2] **Deepak Gopinath** [3] **Guy Rosman** [3]

Abstract

Skill atrophy, the gradual decline of human capability under AI assistance, poses a safety risk in shared-control of semi-autonomous systems, where operators may be unable to distinguish their own inputs from autonomous corrections. We propose Proximal State Nudging (PSN), a shared autonomy algorithm that jointly optimizes for skill development and task performance by nudging users toward states estimated to be most learnable. We first show that PSN outperforms existing shared autonomy baselines in balancing student improvement in unassisted reward with overall shared performance, using simulated students in the classic LunarLander environment. We then present, to the best of our knowledge, the first human subject studies of a planner incorporating learning-compatible shared autonomy: across two driving tasks in the CARLA simulator (High Performance Racing and Parallel Parking, n = 60), PSN produces up to 7x larger gains in unassisted skill than standard blended shared autonomy, while incurring 50% fewer collisions than unassisted self-practice.

1. Introduction

Skill atrophy resulting from AI assistance has been documented across a wide range of domains, including surgery (Sarofim, 2024) and transportation (Casner et al., 2016; de Winter et al., 2023a). Despite this growing evidence, relatively little work has focused on designing AI assistance that actively supports continued human skill improvement. Instead, existing discourse frames the problem as a binary choice: either rely on AI for strong task performance, or completely remove AI support to encourage learning (Dell'Aquila et al., 2023). This framing raises the question: *Can we shape AI assistance to support task performance and safety in the moment, while also improving user's long-term skill development?*

Imagine a student pilot learning how to control an aircraft, or a novice operator learning to teleoperate a robotic manipulator. Human control of such tasks matters for two reasons: first, in high-skill domains – performance driving, surgical training, and rehabilitation — human skill is itself the goal, and AI assistance serves to scaffold its acquisition. Second, human operators might still intervene when automation fails in situations like aircraft control or driving (Endsley, 2017). In both cases, while AI assistance can improve task performance and safety, we argue that shared autonomy systems should also account for how assistance affects users' long-term skill retention. Prior works have explored balancing task performance with skill maintenance, but these approaches often make task-specific assumptions (e.g. access to near-optimal Q-values), and are not grounded in principles from cognitive psychology that provide generalizable theories of how assistance impacts human learning and retention (Wada, 2019; Bragg & Brunskill, 2020).

We propose PROXIMAL STATE NUDGING (PSN), a learning-aware assistance framework that explicitly accounts for its influence on human skill development. PSN is inspired by the Zone of Proximal Development (ZPD (Vygotsky, 1978)) from educational psychology and aims to estimate the "learnability" of states—how conducive they are to supporting user learning—across the environment. For example, in a high performance racing task, a student driver on the racing line might carry a turn cleanly but struggle to recover after drifting wide, making off-line states the most fruitful for skill growth (Figure 1). PSN optimizes over future actions to balance coverage of such high-learnability states with standard shared autonomy objectives, including safety and task performance.

Crucially, PSN estimates learnability without requiring an explicit model of how assistance changes human learning or task-specific assumptions such as access to near-optimal Q-values. Instead, it draws on the ZPD principle that learning is most productive when users face challenges matched to the boundary of their current ability, and operationalizes this by comparing a user's estimated expected reward from

[1]Stanford University, Stanford, CA, USA [2]UCLA, Los Angeles, CA, USA [3]Toyota Research Institute, Los Altos, CA, USA. Correspondence to: Megha Srivastava <megha@cs.stanford.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

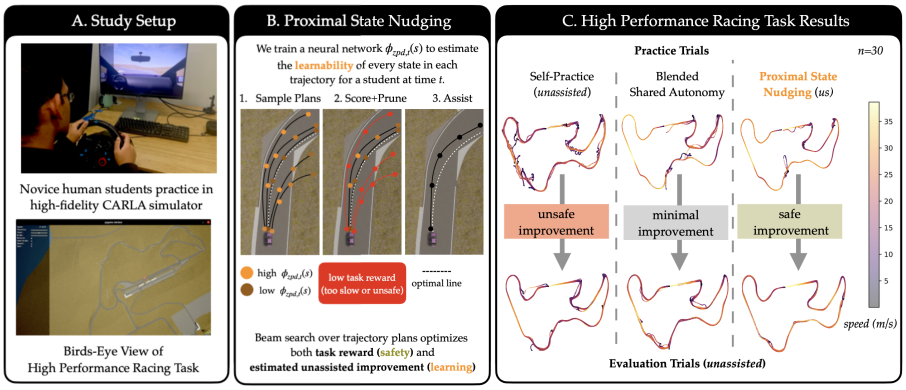


Figure 1: **Overview.** We propose PROXIMAL STATE NUDGING (PSN), a learning-aware shared autonomy algorithm that minimizes human skill atrophy in rapid-control domains, such as High Performance Racing (A). Inspired by the Zone of Proximal Development theory from cognitive psychology (Vygotsky, 1978), PSN maintains an estimator $\phi_{zpd,t}(s)$ of each state's "learnability" – how likely it is to improve a student's unassisted skills – which feeds a beam search-based planner that optimizes for both learning and task reward (B). When used during student practice, PSN eliminates the tradeoff between safe performance and learning; it ensures safer trajectories with fewer collisions in comparision to self-practice without any AI assistance, while resulting in a larger improvement in task reward than standard blending-based shared autonomy (C).

a given state with and without assistance. This grounding in cognitive scaffolding theory, combined with compatibility with any underlying shared autonomy policy, allows us to deploy and evaluate PSN with real human students. Our overall contributions include:

1. An algorithm for learning-compatible shared autonomy grounded in cognitive scaffolding
2. Empirical validation showing that PSN improves upon known shared autonomy baselines in the classic `LunarLander` environment with simulated students
3. Two distinct human-subject studies involving real student learners ($n = 60$) in realistic driving tasks (Parallel Parking and High Performance Racing) using the open-source CARLA autonomous driving simulator (Dosovitskiy et al., 2017), showing that PSN results in up to 7x gains in improving unassisted skill performance than standard blended shared autonomy, while also incurring 50% fewer collisions than unassisted self-practice.

We show that by guiding users toward states that foster learning while maintaining strong task performance, PROXIMAL STATE NUDGING can help prevent skill atrophy.

2. Related Works

PROXIMAL STATE NUDGING lies at the intersection of shared autonomy for human-robot interaction and research on skill atrophy under automation. Shared autonomy systems combine human and AI control to support complex task execution while preserving (partial) human agency, typically assuming users are already skilled. (Dragan & Srinivasa, 2013) introduced policy-blending formalisms for shared control, (Javdani et al., 2015) proposed hindsight optimization under goal uncertainty, and (Reddy et al., 2018) relaxed assumptions on goal spaces via model-free deep RL. Related "minimum intervention" frameworks (Broad et al., 2019) similarly preserve human agency but do not reason about learning.

A smaller line of work explicitly targets skill preservation: (Bragg & Brunskill, 2020) introduced Stochastic Q-Bumpers, which, unlike our work, assumes Q-learning updates for both student and expert policies and then modulates assistance based on Q-value gaps to retain learning opportunities. (Fitzsimons et al., 2019) and (Srivastava et al., 2022) modulate assistance for motor skill training; (Agarwal & Deshpande, 2019) used shared autonomy to induce variability in rehabilitation robotics, implicitly targeting high-ZPD states; and (Tian et al., 2023) modeled human learning dynamics in interaction. Recently (Srivastava et al., 2025) applied assistive coaching to high-performance racing also inspired by cognitive scaffolding, but did not incorporate learning objectives explicitly into a shared autonomy planner.

Meanwhile, skill atrophy from automation over-reliance has been documented across safety-critical control domains, including aviation (Casner et al., 2014), driving (de Winter et al., 2023b), and surgery (Sarofim, 2024), with medical evidence that even expert performance degrades once assistance is removed (Parchmann et al., 2024). Our work directly targets this gap, aligning with broader human-

centered AI efforts emphasizing user empowerment (Ellis et al., 2025) while focusing specifically on sustained skill development in cyber-physical shared control.

3. Formalism

We consider a student with an unassisted policy $\pi_{\text{student},t} \in \Pi$ at each *learning timestep* t, mapping states to actions ($s \in S \to a \in A$). Our objective is to design a assistive policy for $\pi_{\text{student},t}$ that satisfies two properties:

1. **Safety:** Safe and effective task completion across time, measured by high reward and task performance averaged across episodes when the student is *assisted*.
2. **Learning:** $\pi_{\text{student},t}$ improves over time, measured by high reward and task performance averaged across episodes when the student is *not assisted*.

As described earlier, most approaches to shared autonomy often fail our second goal, as providing useful assistance might result in the student incorrectly attributing high reward to their own incorrect input. However, determining how to structure assistance that actually promotes student learning remains challenging due to the lack of a strong model of human adaptation, and prior works on modeling human learning make strong assumptions such as using Wiener processes or linear-quadratic dynamics for the student model (Yu et al., 2023; Tian et al., 2023).

In contrast, PROXIMAL STATE NUDGING adopts a state-centric view of skill learning, inspired by Vygotsky's Zone of Proximal Development (ZPD) to adapt instruction difficulty (Vygotsky, 1978). We estimate $\phi_{\text{zpd},t}$, the likehood that a state $s \in S$ supports the student's learning at a given point in time t. States that are too difficult (low chance of high reward) or too easy (already mastered) should have low learnability. The shared policy nudges the student toward states with higher $\phi_{\text{zpd},t}$ by selecting actions that balance task optimality and state learnability. We now expand on PSN.

3.1. Shared Control Planning for Assistance

PROXIMAL STATE NUDGING can build upon any existing shared control strategy $\pi_{\text{shared}}(a|s)$ which, given the student policy $\pi_{\text{student,t}}(a|s)$ and an expert agent policy $\pi_{\text{expert,t}}(a|s)$, selects the next action for a given level of assistance α. For example, a standard form of assistance is blending-based shared autonomy, which samples from a convex combination of the optimal agent and student policies, controlled by α ((Wang et al., 2020; Steele & Gillespie, 2001)):

$$\pi_{\text{shared}}(a \mid s) \triangleq \alpha\, \pi_{\text{expert}}(a \mid s) + (1-\alpha)\, \pi_{\text{student}}(a \mid s), \qquad \alpha \in [0,1]. \tag{1}$$

During planning, PSN modifies π_{shared} by first employing a beam search over a set of length-T sampled trajectories $\mathcal{B}_t(\hat{s}) \sim \pi_{\text{shared}}(\cdot \mid \hat{s})$, starting at the current student's state $\hat{s}$. We then select the first action of the sampled trajectory that maximizes the score J, defined as a weighted sum of average predicted state learnability and task reward:

$$\begin{aligned} J_t(\tau) &\triangleq w_1 \left(\frac{1}{T} \sum_{i=0}^{T-1} \phi_{\text{zpd},t}(s_i) \right) + w_2\, R(\tau), \\ a_{\text{PSN},t}(s) &\triangleq a_0(\tau^\star), \qquad \tau^\star \in \arg\max_{\tau \in \mathcal{B}_t(s)} J_t(\tau). \end{aligned} \tag{2}$$

Here, R is the task reward function, and $\phi_{\text{zpd},t}(s_i)$ is the estimated learnability of state s_i, which we discuss next.

3.2. Proximal State Estimation: Identifying Learnable States

The key mechanism of PROXIMAL STATE NUDGING is estimating a state *learnability* function $\phi_{\text{zpd},t}(s)$, which operationalizes the concept of ZPD, or the gap between what tasks a student can do independently and with assistance, from cognitive psychology (Vygotsky, 1978). Given a current student's state $\hat{s}$, our goal is to estimate whether there is room for improvement in task performance given the student's current skill level, which is naturally upper-bounded by the performance achievable under expert assistance. Accordingly, if we sample trajectories starting from the student's state $\hat{s}$ with (π_{shared}) and without ($\pi_{\text{student},t}$) assistance, we can approximate $\phi_{\text{zpd},t}(\hat{s})$ as the difference in expected task reward:

$$\mathbb{E}_{\tau \sim \pi_{\text{shared}}, s_0 = \hat{s}}[R_{\text{task}}(\tau)] - \mathbb{E}_{\tau \sim \pi_{\text{student},t}, s_0 = \hat{s}}[R_{\text{task}}(\tau)].$$

For efficiency, we reuse the student's training rollouts from s_{start} to train $\phi_{\text{zpd},t}$, treating each visited state along a trajectory as a candidate $\hat{s}$. Any state $\hat{s}$ from which the expected task reward is similar with or without assistance — either because they are too easy or too hard for the student agent — will receive low $\phi_{\text{zpd},t}(\hat{s})$. Furthermore, the $\phi_{\text{zpd},t}$ estimator can be updated at each learning step t (or at a fixed interval), thus evolving with the student policy. In practice, $\phi_{\text{zpd},t}(\hat{s})$ is normalized to be between 0 and 1 via the sigmoid function, and function approximation can be used for large state dimensions. By not requiring any explicit student model, PROXIMAL STATE NUDGING can be used in settings with real human student learners, unlike prior work (Section 5).

3.3. Algorithm Overview and Adaptive α

Algorithm 1 summarizes our full approach for PROXIMAL STATE NUDGING. UpdateZPD trains the $\phi_{zpd,t}(s)$ estimator at time t on states where the student is either assisted and

unassisted for the entire episode; its sample complexity can be reduced by using a small model size. While in simulation UpdatePolicy can be implemented using any standard reinforcement learning algorithm, our primarary goal is to support human learners, where this function is implicit and likely task-dependent. Finally, while we write Algorithm 1 as if the BeamSearch procedure happens at every state during training, in practice this can be implemented at fixed interval to reduce latency.

The hyperparameter α typically denotes a fixed level of assistance in shared autonomy methods (Reddy et al., 2018). However, students can only attribute observed outcomes to their own actions when assistance is low, so we taper α in a way that is proportional to the estimate $\phi_{zpd,t}(s)$ of the current state. This encourages student learning in the high learnable states that PROXIMAL STATE NUDGING pushes the student towards.

4. Empirical Results: Simulated Learners

We first compare PROXIMAL STATE NUDGING against existing shared autonomy baselines, using simulated students for fair comparison. Specifically, we focus on the `Lunar Lander` (continuous state) task studied in prior work, including (Bragg & Brunskill, 2020) and (Reddy et al., 2018). To the best of our knowledge, (Bragg & Brunskill, 2020) is the only prior work that also seeks to improve a student's unassisted performance ((Reddy et al., 2018) and other shared autonomy works focus on improving assisted task performance). However, their proposed method, called Stochastic Q-Bumpers, assumed Q-learning style agents, which might not be applicable in real-world human learning tasks like the ones we study in Section 5.1. Nevertheless, as this is our only known baseline we follow (Bragg & Brunskill, 2020) and train Double-DQN student agents (Hasselt et al., 2016), and then analyze the pareto frontier between assisted and unassisted task performance. These two metrics correspond to our goal of balance safety and learning: optimizing for strong assisted task performance ensures safety of the student agent, while strong unassisted task performance demonstrates that such assistance does not result in skill atrophy.

4.0.1. TASK OVERVIEW

`LunarLander` (Brockman et al., 2016) is a classic rapid-control task in which an agent must safely land a spacecraft in a 2D environment. The environment has a continuous 8-dimensional state space encoding position, velocity, orientation, angular velocity, and ground contact indicators, and a discrete action space of four engine commands (left, right, main, none). The multi-dimensional reward function is shaped to encourage proximity to the landing pad, low velocities, upright orientation, and successful leg contact, while penalizing fuel usage and crashes. Total rewards of > 200 indicate strong task performance.

Methods We compare five methods:

- **Bragg et. al., 2020 (Baseline)**: We implement the learning-compatible decision support algorithm from (Bragg & Brunskill, 2020), which overrides the student's action a_{student} with the expert action $a_{\text{expert}} = \arg\max_a Q^*(s,a)$ whenever the Q-value gap $\Delta Q(s) = Q^*(s, a_{\text{expert}}) - Q^*(s, a_{\text{student}})$ exceeds a threshold, with override probability proportional to $\Delta Q(s)$.
- **Reddy et. al., 2018 (Baseline)**: We implement the shared autonomy algorithm introduced in (Reddy et al., 2018), which uses model-free deep RL to train an assistive policy π_{shared} that maximizes task reward while constraining its output to remain close to the student's input.
- **Blended Shared Autonomy (Baseline)**: We implement the standard policy-blending formalism from Equation 1. This baseline is also used in prior work to optimize for assisted task performance, but does not explicitly account for student learning (Dragan & Srinivasa, 2013).
- **No AI Assistance (Baseline)**: $\pi_{student}$ is updated from experience without any form of assistance at any point.
- **Proximal State Nudging (PSN)**: We collect unassisted and assisted rollouts from $\pi_{student}$ every 30 episodes, which are used for both evaluation and updating $\phi_{\text{zpd},t}$. Beam search over $B = 3, T = 3$ trajectories is sampled from the combined $\pi_{student}$ and π_{expert} Q-values. We set R to the task reward, and $w_1 = 0.5$ and $w_2 = 0.5$.

4.0.2. TECHNICAL IMPLEMENTATION

Assistive Control Design For all methods, we use Double DQN to train a 3-layer MLP with hidden-layer size 128 for both π_{expert} and $\pi_{student}$ (Hasselt et al., 2016). We train π_{expert} to convergence for 1000 epochs at a learning rate of 0.001. We report results for assistance level $\alpha = 0.1$, as we observe that higher values do not increase assistive performance, and simply deteriorate the unassisted reward of $\pi_{student}$.

Proximal State Estimation We use a small 3-layer MLP, with hidden layer size 16, for $\phi_{zpd,t}$. We update $\phi_{zpd,t}$ every 30 episodes, how which states are "learnable" for the student agent will change over time.To train $\phi_{zpd,t}$, we first train two separate MLPs (ρ_{sa} and ρ_{ua}), corresponding

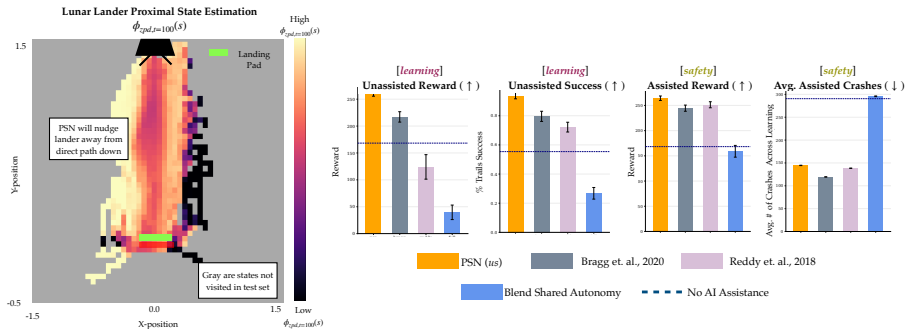


Figure 2: **Lunar Lander Task and Results** (Left) Heatmap showing estimated proximality scores for states in `Lunar Lander`, estimated over states drawn from a test set after training for 100 episodes. States directly below the lander (too easy) or too far to the right (too challenging) are estimated to be less "learnable"; a student agent will face similar expected reward regardless of receiving or not receiving assistance. PSN will nudge the student to states on the left, while maintaining overall task performance. (Right) PSN outperforms all shared autonomy baselines and the no-assistance baseline in learning (high unassisted reward and % successful landings) while maintaining safety (high assisted reward and low total number of crashes during learning). Error bars show standard error across 10 random seeds and 100 episodes.

each to the sampled assisted and unassisted episodes for evaluation, and then define the overall estimator using an affine normalization of the output difference (ρ_{sa} - ρ_{ua}). This two-model approach is necessary due to the continuous state space result in trajectories that rarely overlap exactly.

Figure 2 (left) shows a heatmap of the outputs of $\phi_{zpd,t=3}$ across different states in evaluation rollouts in the `Lunar Environment`. At $t = 3$, or 90 episodes, we observe that $\phi_{zpd,t=3}$ is low in the region directly below the lander's initial state, suggest that the region is "too easy" for the student. Likewise, states at the fringe edges (e.g. far right) have extremely low $\phi_{zpd,t=3}$ estimates, corresponding to challenging regions. At this point in time PSN will likely nudge to student to the middle regions that are off-center; challenging enough to improve learning, yet safe enough to recover from during learning.

4.0.3. Empirical Results

We compare all three methods at the end of training for 300 episodes, which is sufficient time for a student policy without assistance to start to observe successful landings. Figure 2 shows that overall unassisted task reward and number of succesful landings is significantly higher with PSN than all baselines, including the no AI assistance method (navy dashed line) which learns slowly. At the same time, PSN also mantains high assisted reward that is comparable to the Bragg et. al., 2020 and Reddy et. al., 2018 shared autonomy baselines. As expected, the no AI assistance baseline results in a significant number of crashes during learning, which PSN successfully mitigates.

In addition to ensuring a successful balance of both learning and safety for the student agent, PSN holds another advantage over shared autonomy baselines: it does not assume Q-learning style updates from either the student or the expert, nor requires access to the student internal policy (e.g.Q-values). *What happens when we use* PSN *with real human students, who may not follow a straightforward learning strategy?*

5. Empirical Results: Human Subjects

We next evaluate how Proximal State Nudging supports real human users learning two dynamical control tasks: Parallel Parking and High Performance Racing. We built environments for both tasks using the CARLA simulator (Dosovitskiy et al., 2017), often used for autonomous driving research (Li et al., 2023; Srivastava et al., 2025; Costa et al., 2025; Mihaylova et al., 2025). We ask two research questions:

1. **RQ1 (Learning)** Does Proximal State Nudging improve learning (change in unassisted task performance)? We hypothesize that PSN will outperform *blended shared autonomy*, and comparable to a *self-practice* baseline.
2. **RQ2 (Safety)** Does Proximal State Nudging maintain strong overall task performance during learning? We hypothesize that PSN will outperform *self-practice*, while stay comparable to *blended shared autonomy*.

Our unifying hypothesis is that PSN strikes a balance between learning gains (unassisted) and safe task performance during learning (assisted) when compared to baselines. Because both tasks are sufficiently complex that learning expert policies with RL is challenging, the only baseline from Section 4 that we can compare PSN against in this section is

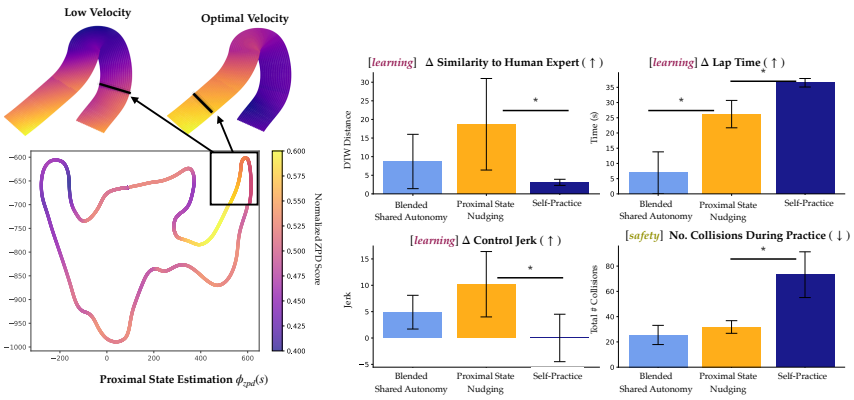


Figure 3: **High Performance Racing Results** (Left) Heatmap $\phi_{\text{zpd, t=1}}$ estimates shows that assistance is predicted to most strongly support student learning at states near high-speed tight turns, and that predicted learnability is sensitive to both location and velocity. (Right) PROXIMAL STATE NUDGING balances the trade-off between learning gains (e.g. improved similarity to expert, lap time, and control jerk) and safety during practice, in comparison to both blended shared autonomy and unassisted self-practice baselines. Bars report mean standard error, stat. sig. determined with Welch's t-test.

the Blended Shared Autonomy baseline. Furthermore, as we anticipate that learning is a slow temporal process for human students, we build our ϕ_{zpd} estimator using offline data, assuming we can treat state learnability as fixed for the 1 hour duration of our experiments. We recruit $n = 60$ participants, mostly university students, across two institutions. Our study was approved by our Institutional Review Board.

5.1. High Performance Racing

In the high-performance racing (HPR) task, users are instructed to drive around a fixed race track with the goal of minimizing their overall lap time. This task is particularly well-suited for studying human skill acquisition because it has clear success metrics for both learning (e.g. improved lap time) and safety (e.g. total number of collisions).

5.1.1. TASK OVERVIEW

We follow the HPR task set-up studied in several prior works (DeCastro et al., 2024; Lidard et al., 2024; Srivastava et al., 2025; Sumner et al., 2025). Specifically, we simulate the 2-mile track in Thunderhill Raceway Park, California in the CARLA simulator, using a vehicle configured for high performance racing. Each state in the HPR task consists of the vehicle's position (x, y, z), angle $(yaw, pitch, roll)$, and speed (vx, vy, vz). The continuous action space consists of steering angle, throttle, and brake control. Each episode (a single lap around the raceway) is evaluated with three metrics: lap time, Dynamic Time Warp distance from an expert demonstration, and smoothness (measured as jerk action input).

Methods We compare three methods:

- **Blended Shared Autonomy (Baseline)**: We blend the inputs of $\pi_{student}$, the user, and π_{expert}, built from an expert human demonstration, with $\alpha = 0.8$ following Equation 1.
- **Proximal State Nudging (PSN)**: We sample, at fixed intervals along the track, the nearest point for each of $B = 5$ distinct human expert trajectories, which cover different near-optimal task strategies. We then set the point in each trajectory that crosses the start of the next interval as the goal for CARLA's Global Route Planner (GRP), which returns a set of waypoint trajectories $\mathcal{B}_t(s)$. Following Equation 2, we set R to the inverse trajectory length (encouraging speed), and $w_1 = 0.5$ and $w_2 = 0.5$.
- **Unassisted Self-Practice (Baseline)**: The student repeatedly practices high performance racing around the track with full control of the vehicle.

5.1.2. TECHNICAL IMPLEMENTATION

Assistive Control Design The assistive control for both Blended Shared Autonomy and PSN use human expert demonstrations provided by (Srivastava et al., 2025). Given a point in an expert trajectory, the Path Planner in the CARLA simulator generates waypoints to input into a PID controller with the driver's state, and the control outputs are used as the actions for π_{expert}.

Proximal State Estimation To avoid introducing latency, we used a Linear Regression model for ϕ_{zpd}. This is suitable as the state space is low-dimensional. To train ϕ_{zpd}, we collect an offline dataset of 50 demonstrations of novice drivers both with and without assistance (at $\alpha = 0.8$). We

follow the same two-model approach, based on assisted and unassisted trajectories, described in Section 4.

Figure 3 (left) shows a heatmap of the outputs of ϕ_{zpd} across different states across the Thunderhill Raceway track, showing that the states with the highest estimated learning potential for students are concentrated at the tight turn towards the end of the track. Furthermore, $\phi_{zpd}(s)$ is very sensitive to the velocity values in s, suggesting that the student's speed affects its learnability. Interestingly, $\phi_{zpd}(s)$ changes laterally across the track, and PSN would nudge a slower student to take the tight turn widely, while a faster student would be encouraged to focus on states after the turn's apex.

5.1.3. Human Subject Study Results

We compare all three methods with the same sequence of trials:

1. 2 **baseline trials** with unassisted user control.
2. 2 **practice trials** with either (i) Blended Shared Autonomy, (ii) Unassisted Self-Practice or (iii) PSN.
3. 2 **evaluation trials** with unassisted user control.

In each trial, subjects were instructed to complete one lap as quickly as possible (under 3 minutes). We recruited 30 students, evenly split between methods. We compute the three learning metrics (lap time, Dynamic Time Wrap distance to a human expert, and control input jerk) as the difference between average baseline and evaluation performance. For safety, we measure the total number of collisions during practice trials, with lower values indicating safer learning.

Our results, reported in Figure 3, show that Proximal State Nudging outperforms Blended Shared Autonomy across all three learning metrics, with a statistically significant difference in lap time ($\sim$ 5x), supporting our hypothesis for **RQ1**. PSN was also significantly stronger than Unassisted Self-Practice for both similarity to expert and jerk metrics, though self-practice led to a significantly stronger improvement in lap time. Finally, users assigned either PSN or Blended Shared Autonomy had low number of collisions during practice, and significantly lower than Unassisted Self-Practice ($< 50\%$), validating our hypothesis for **RQ2**.

One reason why self-practice leads to a significant improvement in lap time, but not other metrics, might be that students are fixated on improving the lap time and ignore other aspects of good driving, like smooth control. In fact, participants assigned PSN during practice noted learning specific turning skills (e.g., *"I learned there are certain chokepoints where you need to reduce speed to avoid spinning out"* and *"learning when to ease off the gas pedal when heading into turns"*),. Furthermore, self-practice students experienced a high number of collisions during practice. These results support our hypotheses that PSN can improve student learning (**RQ1**) while still providing enough assistance to ensure safety (**RQ2**).

5.2. Parallel Parking

Parallel parking is another challenging driving task requiring precise spatial reasoning and planning, often studied due to its difficulty for human drivers and clear success criteria (Chai et al., 2023). The task consists of an approach phase followed by a parking phase, and we assume learning difficulties primarily arise from variation in the endpoint of the approach phase (i.e., the start of the parking phase).

5.2.1. Task Overview

We again use the CARLA simulator, this time building a parallel parking scenario in the provided Town15 multilane environment. Each state includes vehicle position (x, y, z), orientation, and velocity, while continuous actions consist of steering, throttle, and brake. We designated a 13 m gap between two stationary vehicles as the target spot, and each episode (a single parking attempt) is evaluated with two metrics: parking time and number of collisions.

Methods We compare three methods:

- **Blended Shared Autonomy (Baseline):** We blend $\pi_{student}$, the user's input controls, and π_{agent}, a pure-pursuit controller that takes a goal state as input, with $\alpha = 0.8$ following Equation 1.
- **Proximal State Nudging (PSN)** : We sample $B = 8$ different states for the start of the parking phase, and we set the state with the highest ϕ_{zpd} estimate as the goal for a pure pursuit controller that guides the student. Following Equation 2, we set R to a fixed constant, and $w_1 = .5$ and $w_2 = .5$.
- **Unassisted Self-Practice (Baseline)**: The student repeatedly practices parking with full control of the car.

5.2.2. Technical Implementation

Assistive Control Design For our assistive control, π_{expert} uses a non-linear Model Predictive Control (NMPC) policy: a proportional heading-and-speed controller for the approach, followed by a finite-horizon nonlinear program (solved with IPOPT) for parking (Wächter & Biegler, 2006). The NMPC controls are blended with human inputs using $\alpha = 0.8$, and the agent is queried every 8 frames for tractability.

Proximal State Estimation We discretize the high-level state space near the parking spot and estimate ZPD scores for each state. We collect 272 trajectories from 17 novice drivers, each attempting 4 trials from 8 spots (2 assisted, 2 unassisted, randomized). Following Section 3, $\phi_{zpd}(s)$ is

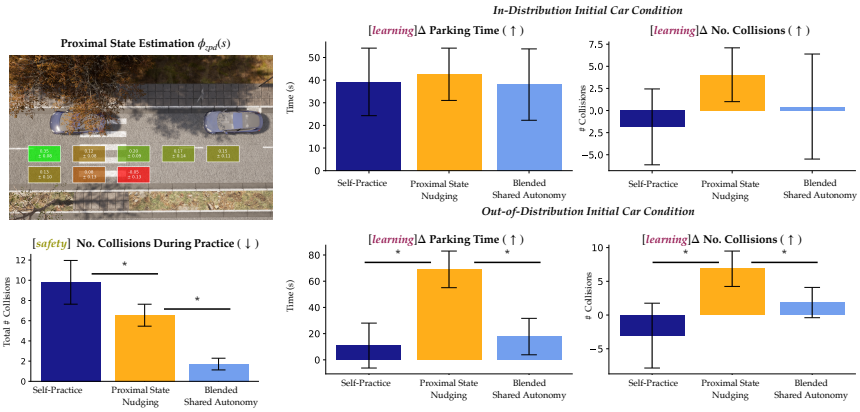


Figure 4: **Parallel Parking Results** (Top Left) Estimated learnability $\phi_{\text{zpd},t=1}$ over discretized approach states near the target parking spot; green indicates higher. (Bar Plots) PROXIMAL STATE NUDGING balances the trade-off between learning gains (e.g. improved time and collisions) and safety (total # of collisions during practice). in comparison to both blended shared autonomy and unassisted self-practice. Bars show mean standard error, stat. sig. via two-sample Welch's t-tests.

computed as the average difference of $\rho_{sa}(s)$ and $\rho_{ua}(s)$, which denote average parking time for assisted and unassisted trials . Figure 4 (left) shows the state directly in front of the target spot has the highest estimated learnability.

5.2.3. HUMAN SUBJECT STUDY RESULTS

We compare all three methods with the same trial order:

1. 2 **baseline trials** with unassisted user control.
2. 4 **practice trials** with either (i) Blended Shared Autonomy, (ii) Unassisted Self-Practice, or (iii) PROXIMAL STATE NUDGING.
3. 2 **evaluation trials** with unassisted user control (in distribution starting angle)
4. 2 **evaluation trials** with unassisted control (out of distribution starting angle not seen during baseline trials)

In each trial, participants were instructed to park as quickly as possible while avoiding collisions. We recruited 30 students, evenly split between conditions. We compute two learning metrics (parking time and # of collisions) as the difference between average baseline and evaluation performance, so larger values indicate stronger learning gains. For safety, we measure the total number of collisions during practice trials, with lower values indicate safer learning. Our results (see Figure 4), show that PROXIMAL STATE NUDGING yields greater improvement than both unassisted self-practice and blended shared autonomy in improving parking time and reducing collisions, supporting our hypothesis for **RQ1**. However, statistically strong effects were primarily observed for the randomized-angle out-of-distribution starting angle ($>$ 7x), which suggests that PSN's effect is more salient in conditions that are unfamiliar to a student. Furthermore, users assigned PSN had significantly fewer collisions during the practice trials than Unassisted Self-Practice ($<$ 60%) which validates our hypothesis for **RQ2**, although Blended Shared Autonomy resulted in an even more significantly safer practice. Participants who practices without assistance mainly reported self-learning steering wheel sensitivity, while PSN participants more often described learning new parking strategies (e.g., *"I came to better understand where I should turn my wheel."*). Overall, the results for our Parallel Parking task support our prior experimental results in affirming the goal of PROXIMAL STATE NUDGING to provide assistance that can ensure safety without compromising users' learning.

6. Discussion and Future Work

Across simulated agents in `LunarLander` and $n = 60$ human participants in two CARLA driving tasks, PROXIMAL STATE NUDGING resolved the assistance–learning trade-off incurred by prior shared autonomy methods: up to 7$\times$ stronger unassisted skill gains than blended shared autonomy, with 50% fewer collisions than self-practice. To our knowledge, these are the first human-subject results for learning-compatible shared autonomy. Like all shared autonomy methods (Dragan & Srinivasa, 2013; Reddy et al., 2018), PSN assumes access to an expert policy π_{expert}; we inherit no stronger requirement than the baselines we compare against. Jointly learning π_{expert} alongside $\phi_{\text{zpd},t}$, or updating $\phi_{\text{zpd},t}$ online during deployment (as we do in simulation), are natural next steps, as are longer studies to further test long-term retention. While we focus on rapid cyber-physical control where skill atrophy is most safety-critical, potential extensions can include cognitive domains such as AI-assisted coding (Perry* et al., 2023), where assistance granularity (a token, a line, a function) provides a natural mechanism for PSN.

References

Agarwal, P. and Deshpande, A. A framework for adaptation of training task, assistance and feedback for optimizing motor (re)-learning with a robotic exoskeleton. *IEEE Robotics and Automation Letters*, 4:808–815, 2019. doi: 10.1109/LRA.2019.2891431.

Bragg, J. and Brunskill, E. Fake it till you make it: Learning-compatible performance support. In Adams, R. P. and Gogate, V. (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 915–924. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/bragg20a.html.

Broad, A., Murphey, T., and Argall, B. Highly parallelized data-driven mpc for minimal intervention shared control. In *Robotics: science and systems*, 2019.

Brockman, G., Cheung, V., Pettersson, L., et al. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016.

Casner, S. M., Hutchins, E. L., and Norman, D. The impact of automation on pilot cognition and performance. *Human Factors*, 56(1):1–17, 2014.

Casner, S. M., Hutchins, E. L., and Norman, D. The challenges of partially automated driving. *Communications of the ACM*, 59(5):70–77, 2016.

Chai, R., Liu, D., Liu, T., Tsourdos, A., Xia, Y., and Chai, S. Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver. *IEEE Transactions on Automation Science and Engineering*, 20 (3):1633–1647, 2023. doi: 10.1109/TASE.2022.3183610.

Costa, J., Morgan, A., Yasuda, H., Sumner, E. S., Gopinath, D., Chau, S., Nguyen, H., Best, A., Rosman, G., and Chen, T. L. From dashboards to dialogue: Evaluating a conversational ai coach for performance driving skill development. In *Proceedings of the 17th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '25, pp. 9–18, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720130. doi: 10.1145/3744333.3747811. URL https://doi.org/10.1145/3744333.3747811.

de Winter, J. C., Petermeijer, S. M., and Abbink, D. A. Shared control versus traded control in driving: a debate around automation pitfalls. *Ergonomics*, 66(10):1494–1520, 2023a.

de Winter, J. C. F., Petermeijer, S. M., and Abbink, D. A. Shared control versus traded control in driving: a debate around automation pitfalls. *Ergonomics*, 66(10):1494–1520, October 2023b.

DeCastro, J., Silva, A., Gopinath, D., Sumner, E., Balch, T. M., Dees, L., and Rosman, G. Dreaming to assist: Learning to align with human objectives for shared control in high-speed racing, 2024. URL https://arxiv.org/abs/2410.10062.

Dell'Aquila, E., Rossi, F., and Ronchetti, M. Chatgpt in education: transforming digital learning and assessment. *Applied Sciences*, 13(18):10053, 2023.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, pp. 1–16, 2017.

Dragan, A. D. and Srinivasa, S. S. A policy-blending formalism for shared control. *International Journal of Robotics Research (IJRR)*, 32:790–805, 2013.

Ellis, E., Myers, V., Tuyls, J., Levine, S., Dragan, A., and Eysenbach, B. Training llm agents to empower humans. *arXiv preprint arXiv:2510.13709*, 2025.

Endsley, M. R. Autonomous driving systems: A preliminary naturalistic study of the tesla model s. *Journal of Cognitive Engineering and Decision Making*, 11(3):225–238, 2017.

Fitzsimons, K., Kalinowska, A., Dewald, J., and Murphey, T. Task-based hybrid shared control for training through forceful interaction. *The International Journal of Robotics Research*, 39:1138 – 1154, 2019. doi: 10.1177/0278364920933654.

Hasselt, H. v., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100. AAAI Press, 2016.

Javdani, S., Srinivasa, S., and Bagnell, J. A. Shared autonomy via hindsight optimization. In *Robotics: Science and Systems*, 2015.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, Z., Sun, Y., Tomizuka, M., and Zhan, W. Learning from active human involvement through proxy value propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Lidard, J., Hu, H., Hancock, A., Zhang, Z., Contreras, A. G., Modi, V., DeCastro, J., Gopinath, D., Rosman, G., Leonard, N. E., et al. Blending data-driven priors in dynamic games. *arXiv preprint arXiv:2402.14174*, 2024.

Mihaylova, T., Reitmann, S., Topp, E. A., and Kyrki, V. Injecting conflict situations in autonomous driving simulation using carla. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1052–1056, 2025. doi: 10.1109/HRI61500.2025.10974213.

Parchmann, F. et al. Ai-induced deskilling in medicine: A mixed-method review and research agenda for healthcare and beyond. *Artificial Intelligence Review*, 2024.

Perry*, N., Srivastava*, M., Kumar, D., and Boneh, D. Do users write more insecure code with ai assistants? In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2023.

Reddy, S., Dragan, A. D., and Levine, S. Shared autonomy via deep reinforcement learning. In *Arxiv 1802.01744*, 2018. URL https://arxiv.org/abs/1802.01744.

Sarofim, M. Devil's advocate: exploring the potential negative impacts of artificial intelligence on the field of surgery. *Journal of Medical Artificial Intelligence*, 7, 2024.

Srivastava, M., Biyik, E., Mirchandani, S., Goodman, N., and Sadigh, D. Assistive teaching of motor control tasks to humans. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 28517–28529. Curran Associates, Inc., 2022.

Srivastava, M., Iranmanesh, R., Cui, Y., Gopinath, D., Sumner, E. S., Silva, A., Dees, L., Rosman, G., and Sadigh, D. Shared autonomy for proximal teaching. In *Proceedings of the 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2025)*, pp. 232–241, 2025.

Steele, M. and Gillespie, R. B. Shared control between human and machine: Using a haptic steering wheel to aid in land vehicle guidance. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 45(23):1671–1675, 2001. doi: 10.1177/154193120104502323. URL https://doi.org/10.1177/154193120104502323.

Sumner, E., Gopinath, D. E., Dees, L., Gomez, P. R., Cui, X., Silva, A., Costa, J., Morgan, A., Schrum, M., Chen, T. L., Balachandran, A., and Rosman, G. Simcoachcorpus: A naturalistic dataset with language and trajectories for embodied teaching, 2025. URL https://arxiv.org/abs/2509.14548.

Tian, R., Bajcsy, A., Tomizuka, M., and Dragan, A. D. Towards modeling and influencing the dynamics of human learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 12–21. Association for Computing Machinery, 2023. ISBN 978-1-4503-9973-7. doi: 10.1145/3568162.3578629.

Vygotsky, L. S. *Mind in society: The development of higher psychological processes*, volume 86. Harvard university press, 1978.

Wächter, A. and Biegler, L. T. On the implementation of a primal-dual interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi: 10.1007/s10107-004-0559-y.

Wada, T. Simultaneous achievement of driver assistance and skill development in shared and cooperative controls. *Cognition, Technology & Work*, 21(4):631–642, 2019.

Wang, W., Na, X., Cao, D., Gong, J., Xi, J., Xing, Y., and Wang", F.-Y. Decision-making in driver-automation shared control: A review and perspectives. *IEEE/CAA Journal of Automatica Sinica*, 7(JAS-2020-0177):1289, 2020. ISSN 2329-9266. doi: 10.1109/JAS.2020.1003294.

Yu, C., Xu, Y., Li, L., and Hsu, D. Coach: Cooperative robot teaching. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1092–1103. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/yu23b.html.

A. Appendix

A.1. Algorithm

Algorithm 1 Proximal State Nudging

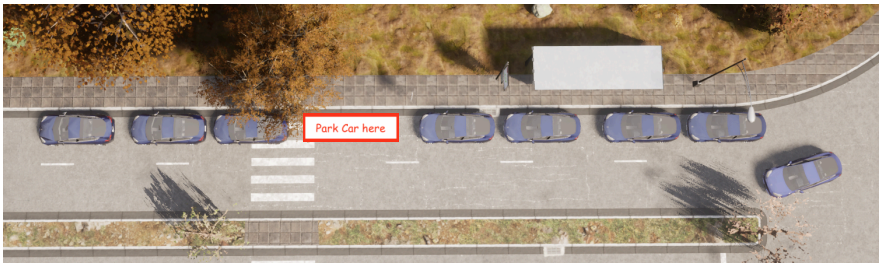
Input: Task environment with state space $\mathcal{S}$, action space $\mathcal{A}$
  Expert policy π_{expert}
  Base shared policy π_{shared}
  Assistance level $\alpha \in [0, 1]$
  Scoring weights $w_1, w_2 \in [0, 1]$
  Planning horizon T
  Beam search size B
  Task reward function R_{task}
Initialize: Student policy $\pi_{\text{student},0}$, empty datasets $\mathcal{D}_{\text{shared}}$ and $\mathcal{D}_{\text{student}}$, proximal state estimator $\phi_{\text{zpd},t}$
for learning timestep t **do**
  // Shared control execution
  for each encountered state s **do**
    $\alpha' \leftarrow \alpha * (1 - \phi_{\text{zpd},t}(s))$
    $\mathcal{B}_t(s) \leftarrow \text{BeamSearch}(\pi_{\text{shared}(a|s,\alpha')}, s, T, B)$
    for each trajectory $\tau = (s_0, a_0, \ldots, s_{T-1}) \in \mathcal{B}_t(s)$ **do**
      $J_t(\tau) \leftarrow w_1 \left(\frac{1}{T} \sum_{i=0}^{T-1} \phi_{\text{zpd},t}(s_i) \right) + w_2 R_{\text{task}}(\tau)$
    end for
    $\tau^\star \leftarrow \arg\max_{\tau \in \mathcal{B}_t(s)} J_t(\tau)$
    $a_{\text{PSN},t}(s) \leftarrow a_0(\tau^\star)$
    Add $(s, a_{\text{PSN},t}(s), r, s')$ to student experience
  end for
  // Update proximal state estimator
  Collect assisted episodes: $\mathcal{D}_{\text{shared}} \sim \pi_{\text{shared}}$
  Collect unassisted episodes: $\mathcal{D}_{\text{student}} \sim \pi_{\text{student},t}$
  $\phi_{\text{zpd},t} \leftarrow \text{UpdateZPD}(\mathcal{D}_{\text{shared}}, \mathcal{D}_{\text{student}})$
  // Update student policy
  $\pi_{\text{student},t+1} \leftarrow \text{UpdatePolicy}(\pi_{\text{student},t}, \text{experience})$
end for

A.2. User Study Instructions

A.3. `LunarLander` Training Details

π_{agent} is a pre-trained DQN-based policy trained for 100 episodes, with learning rate .001, and an MLP-archicture consisting of 2 linear layers and hidden dimension 64. ϕ_{zpd} is also an MLP with 3 linear layers of dimensions 32, 32, and 16 respectively. All evaluation results in the paper are averaged across 100 trials.

Thank you for participating in our user study! Your goal is to control a car such that it is parked between the two stationary cars on the curb ("parallel parking"). You will be able to control the car's wheel for steering, and pedals for brake and throttle. A bird's eye view is below:



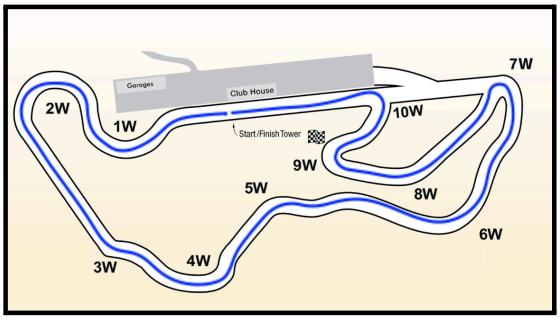
You will complete 10 trials of this task. In particular, ***trials 2-4 will include a moderate amount of AI assistance to help you with parking.*** You will be asked to attempt the following (in order):

1. You will be given **2 minutes** to familiarize yourself with the car's controls in a random environment. Familiarize yourself with steering, throttle, braking, and reversing.
2. **2 baseline** trials where you will have full control. The starting angle of the car will be straight, in the direction of parking.
3. **4 practice** trials where you will receive a moderate amount of AI assistance to help you with parking. The AI assistance might nudge you to different parts of the environment, but your control inputs will still affect the vehicle's behavior. The starting angle of the car will be straight, in the direction of parking.
4. **2 evaluation** trials where you will have full control, and the starting angle of the car will be straight, in the direction of parking.
5. **2 evaluation** trials where you will have full control, and the starting angle of the car will be randomized in order to create a more challenging test environment.

The simulation for each trial will end once you park the car properly (e.g. straight angle and exactly between the two cars). There is a 3 minute time limit for each trial.

Figure 5: User study instructions for parallel parking task.

Thank you for participating in our user study! Your goal is to drive a racecar in a simulated environment of the Thunderhill race track in **as little time as possible without crashing**.. A bird's eye view of the track, with an expert's path in blue, is below:



You will complete 6 trials of this task. In particular, ***trials 3-4 will include a moderate amount of AI assistance to help you with racing.*** You will be asked to attempt the following (in order):

1. **2 baseline** trials where you will have full control of the car.
2. **2 practice** trials where you will receive a moderate amount of AI assistance to help you with racing. The AI assistance might nudge you to different parts of the track you were not planning to go to, but your control inputs will still affect the vehicle's behavior.
3. **2 evaluation** trials where you will have full control of the car.

The simulation for each trial will end once you pass the finish line on the track (shown in red). ***Please turn left on all forks (there are barriers to help guide you).***

Figure 6: User study instructions for HPR task.