PEDRA: PErsonalized Driving Representation for Autonomy

Boyang Tian
Department of Computer & Information Sciences
University of Delaware
Newark, USA
tby@udel.edu

Weisong Shi
Department of Computer & Information Sciences
University of Delaware
Newark, USA
weisong@udel.edu

***Abstract*—Personalizing autonomous vehicles to individual drivers requires translating behavioral preferences into constraints that motion planners can directly enforce. Existing methods either learn driver embeddings that lack control semantics or train driver-specific policies that sacrifice interpretability and require costly retraining. We propose encoding personalization as a personalized safety envelope—four directional margins (front, back, left, right) that capture driver-specific spacing habits while maintaining RSS-consistent safety guarantees. We present PEDRA (PErsonalized Driving Representation for Autonomy), a lightweight framework that learns this geometric representation from short driving sessions and deploys it in real time. A multi-scale temporal encoder captures driving habits across immediate reactions (1–2 s), tactical maneuvers (5–10 s), and sustained patterns (20–40 s); a context-aware decoder then modulates margins using the Context-aware Risk Index (CRI), enabling immediate integration with existing planners. Evaluated on CARLA's Bench2Drive benchmark (220 accident-prone routes) across three risk-tolerance regimes (High-RT, Moderate-RT, Low-RT), PEDRA monotonically reduces RSS violations by 40% and improves habit fidelity from 0.72 to 0.87 as personalization data grows from 5 to 15 minutes. On the high-risk route subset, the Low-RT profile achieves a 35% lower collision rate and a 74% lower RSS violation rate (0.073 vs. 0.285) than an embedding-based baseline, while the High-RT profile reproduces assertive driving patterns—confirming that PEDRA faithfully encodes individual style rather than converging to a single safe policy. The system operates at approximately 9.9 Hz with only 6.4 ms personalization overhead per frame.**

***Index Terms*—User-Centric Intelligent Vehicle Technologies; Collision Avoidance Algorithms; Adaptive Vehicle Control; Autonomous Driving**

I. Introduction

Drivers differ fundamentally in how they perceive risk, manage following distance, and respond to dynamic traffic—differences that a one-size-fits-all autonomous vehicle ignores at the cost of comfort, trust, and adoption. Personalized autonomous driving aims to align vehicle behavior with individual driver preferences while preserving safety guarantees. A growing body of work addresses this challenge through driver embedding, imitation learning, or rule-based safety frameworks [1]–[3]. While these methods capture aspects of driver identity or provide formal guarantees, they typically produce representations that are either non-actionable for real-time control or agnostic to individual spacing habits. A driver who habitually maintains large following distances or avoids tight lane changes cannot express these preferences through a latent embedding alone—the embedding must be re-interpreted by a downstream planner that was never designed to consume it. This leaves a critical gap: *how can driver-specific behavior be encoded as geometric constraints that motion planners directly consume?*

We address this gap by introducing the **personalized safety envelope**—a compact representation that encodes driver-specific spacing habits as four directional margins (front, back, left, right). Unlike latent embeddings that require downstream interpretation, this geometric form serves as explicit planning constraints while maintaining RSS-consistent safety bounds. We present **PEDRA** (PErsonalized Driving Representation for Autonomy), a lightweight framework that learns this representation from short driving sessions and deploys it in real time. A multi-scale temporal encoder distills ego-centric trajectory windows into a habit vector; a context-aware decoder then modulates the four margins using the Context-aware Risk Index (CRI) [4]—a directional risk signal derived from relative motion and right-of-way context—enabling the envelope to expand on threatened sides in proportion to local hazard. Critically, the envelope geometry is directly consumable by existing motion planners without architectural changes, making PEDRA a drop-in personalization layer rather than a replacement for the planning stack. To capture the full spectrum of individual driving preferences, we construct three driver profiles characterized by distinct risk-tolerance regimes—**High-RT**, **Moderate-RT**, and **Low-RT**—derived from quantitative separation across acceleration, jerk, CRI exposure, and danger ratio statistics.

We validate PEDRA on CARLA's Bench2Drive benchmark [5], a challenging evaluation suite of 220 routes specifically constructed to stress-test autonomous driving systems in accident-prone scenarios, demonstrating monotonic safety and fidelity improvement with increasing personalization data and faithful style reproduction across all three risk-tolerance regimes. The system operates at approximately 9.9 Hz with only 6.4 ms personalization overhead per frame.

Our contributions are:

- We formalize the *personalized safety envelope* as a geometric representation that bridges driver behavior and

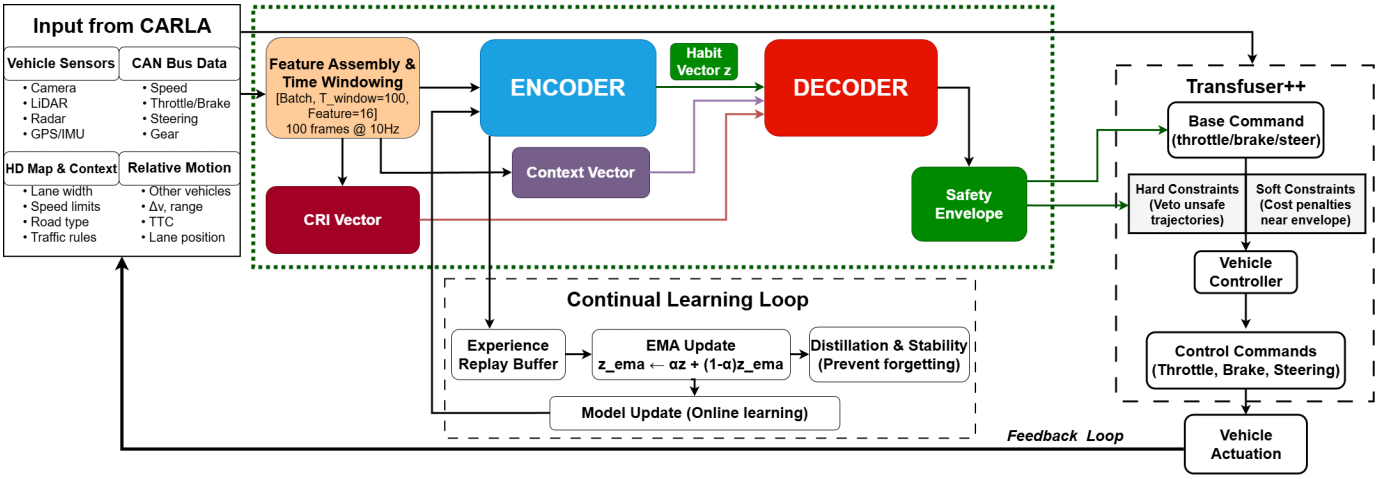

Fig. 1: **PEDRA integration in the driving stack.** CARLA provides vehicle sensor streams, Controller Area Network (CAN) bus signals, HD map/context, and surrounding-agent relative motion. These inputs are assembled and windowed into short feature sequences, then fed to a temporal encoder–decoder that fuses the windowed features, a compact context vector, and a Context-aware Risk Index (CRI) vector (see Sec. III-A) to predict a personalized four-direction safety envelope (front/back/left/right margins). The envelope is smoothed via EMA (exponential moving average), checked for intrusions, and applied as hard/soft constraints to gate a Transfuser++ planner and the low-level controller, which outputs throttle, brake, and steering commands. A continual-learning loop with replay, EMA (exponential moving average)-based stabilization, and distillation updates the habit representation online, closing the feedback loop.

motion planning constraints, enabling direct consumption by existing planners without retraining.

- We present PEDRA, a real-time framework that learns and deploys this representation with RSS-consistent safety bounds and CRI-aware risk modulation [4] from short driving sessions.
- We validate PEDRA on Bench2Drive across 220 routes, demonstrating monotonic data scaling and faithful style reproduction across three risk-tolerance regimes [5].

II. Related Work

Driver Style Representation. A substantial body of work represents driving style as compact embeddings learned from sensor data. Driver2Vec [1] and D-CRNN [6] use temporal convolutions and recurrent networks to encode driver-discriminative patterns from brief intervals, with subsequent work extending to GPS traces, attention mechanisms, and autoencoder-regularized models [7], [8]. Reviews confirm that speed, headway, and steering patterns encode stable individual styles [9]. However, these embeddings primarily serve driver identification or classification tasks and are not directly consumable by motion planners. In contrast, PEDRA learns a geometric representation—four directional margins—that planners can use as explicit spatial constraints without additional interpretation.

Learning-Based Personalized Control. Inverse reinforcement learning (IRL), generative adversarial imitation learning (GAIL), and preference-based reinforcement learning (PbRL) offer principled frameworks for learning user-specific policies from demonstrations [2], [10]–[12]. Recent variants incorporate human feedback and risk-aware objectives [13], [14], improving alignment with user preferences. These methods can capture nuanced driving styles but typically require retraining for each new driver and produce policies whose internal structure is difficult to audit. Our approach instead outputs interpretable, per-axis spacing limits that can be inspected and overridden, enabling rapid adaptation to new drivers without full policy retraining.

Safety Envelopes and Risk Assessment. Responsibility-Sensitive Safety (RSS) [3] and related frameworks [15], [16] provide formal safety guarantees through pre-defined spatial margins. These margins are effective at preventing collisions but are typically fixed across drivers, ignoring individual spacing preferences. Context-aware Risk index [4] extend this line by incorporating environmental factors into directional risk assessment. PEDRA builds on both ideas: we modulate a personalized envelope using directional risk signals while enforcing RSS-consistent lower bounds, combining individual adaptation with formal safety guarantees.

LLM-Conditioned Driving. Recent work explores large language models (LLMs) for style-conditioned planning, mapping textual prompts or user profiles to driving behavior [17]–[20]. While these approaches leverage rich world knowledge, they introduce inference latency and verification challenges that complicate real-time deployment. PEDRA routes personalization through a lightweight temporal encoder operating at frame rate, preserving both responsiveness and auditability.

III. Methodology

PEDRA converts driver behavior into a four-directional safety envelope through three stages: (1) feature windowing and weak supervision to extract spacing habits from driving logs, (2) multi-scale temporal encoding to capture habits across different

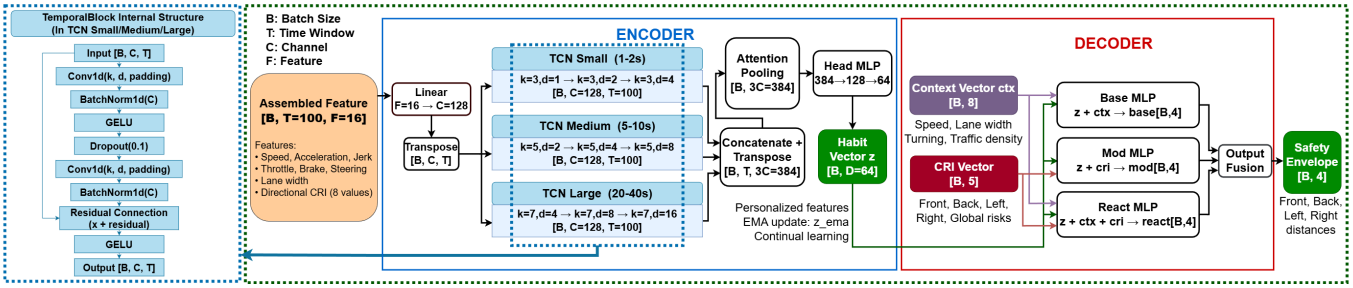

Fig. 2: **PEDRA encoder–decoder architecture with temporal block details.** *Left panel*: Internal structure of the Temporal Convolutional Block used in each TCN branch, showing the residual connection flow (Conv1d–BatchNorm1d–GELU–Dropout–Residual). *Right panel*: Complete multi-scale encoder–decoder pipeline. Assembled features are linearly projected and processed through three parallel TCN branches (Small: 1-2s, Medium: 5-10s, Large: 20-40s) to capture multi-horizon temporal patterns. Outputs are concatenated, attention-pooled, and refined by a Head MLP to produce the habit vector z (green). The decoder fuses z with context vector ctx and CRI vector through three MLP heads (Base/Mod/React) to generate the final four-directional safety envelope (Front, Back, Left, Right distances). Color coding aligns with Fig. 1: blue for encoder, green for habit representation, maroon for risk context, and red for decoder outputs.

time horizons, and (3) context-aware decoding to produce RSS-bounded margins that adapt to real-time risk. Fig. 1 illustrates the complete pipeline.

A. Context-aware Risk Index (CRI)

We first introduce the **Context-aware Risk Index (CRI)** [4], which provides the directional risk signal used throughout PEDRA. The motivation is to make personalization *context-sensitive*: a driver's preferred spacing should be expressed as a geometric constraint under nominal conditions, but the envelope must expand on threatened sides when local hazard increases, while respecting RSS-consistent safety requirements.

At each time step t, CRI outputs a normalized directional risk vector

$$r_t = [r_t^f, r_t^b, r_t^l, r_t^r, r_t^g] \in [0,1]^5,$$

where $r_t^f, r_t^b, r_t^l, r_t^r$ denote front/back/left/right risk levels and r_t^g is a global risk summary. Each component is derived from relative motion cues (e.g., distance and relative speed leading to TTC-like interaction severity) together with lane and right-of-way context (e.g., adjacency, junction/intersection structure), and then normalized to $[0,1]$ with RSS-consistent filtering such that larger values reflect stronger local hazard.

In this work, CRI is not used to directly generate control commands. Instead, r_t is fed into the envelope decoder (Section III) to (i) modulate the learned driver-specific baseline margins and (ii) provide a principled risk prior for expanding the envelope on threatened directions, enabling immediate integration with downstream planning and control.

B. Feature Windowing and Weak Supervision

At each time step t, we assemble a fixed-length window of multi-channel telemetry:

$$X_t = \big[x_{t-T+1}, x_{t-T+2}, \ldots, x_t\big] \in \mathbb{R}^{T\times F},$$

where each frame x_τ concatenates ego kinematics (speed, longitudinal/lateral acceleration, jerk), control traces (throttle, brake, steer, steer-rate), lane context (lane width, adjacency/junction flags), and interaction surrogates (front/side clearances and relative rates). A compact context vector $c_t \in \mathbb{R}^p$ summarizes slowly varying factors (e.g., speed range, road type). In parallel, the same windowed signals feed a lightweight CRI module (Section III-A) that outputs $r_t \in [0,1]^5$ (front/back/left/right/global) for the decoder.

Per-feature online normalization produces a standardized tensor X'_t. Let $\mu \in \mathbb{R}^F$ and $\sigma \in \mathbb{R}^F$ be running means and standard deviations, and $M_t \in \{0,1\}^{T\times F}$ a validity mask. We compute

$$X'_t = M_t \odot \frac{X_t - \mathbf{1}\mu^\top}{\mathbf{1}\sigma^\top} + (1 - M_t) \odot 0.$$

a) Weak supervision.: Rather than requiring manual annotation of preferred spacing, we derive supervision from observed driving behavior. Weak labels for the "space actually used" along four sides $e_t^{\text{act}} = [e^f, e^b, e^l, e^r]^\top \in \mathbb{R}^4$ are extracted from logged trajectories: front/back margins are approximated as minimum longitudinal headway to lead/follow vehicles over the window; left/right margins are minimum lateral clearance to lane boundaries or adjacent vehicles, capped by lane width. An availability mask $m_t \in \{0,1\}^4$ nulls sides without valid observations. This yields low-cost supervision without dense human annotation while keeping targets in physically meaningful ranges.

b) Habit Similarity Index.: To quantify how well the deployed envelope matches a driver's observed spacing preferences at test time, we define the **Habit Similarity Index** (HSI) as the cosine similarity between the predicted envelope vector and the weak-label vector extracted from the corresponding test trajectory:

$$\text{HSI}_t = \frac{\hat{e}_t \cdot e_t^{\text{act}}}{\|\hat{e}_t\|_2 \, \|e_t^{\text{act}}\|_2} \in [0,1], \tag{1}$$

where $\hat{e}_t \in \mathbb{R}^4_{\geq 0}$ is the predicted four-side envelope and $e_t^{\text{act}} \in \mathbb{R}^4$ is the observed spacing extracted from the test

trajectory (front/back headway and left/right clearance, as defined in Section III). HSI is computed exclusively on held-out test routes disjoint from the training data (`longest6` subset), ensuring that the metric reflects generalization to unseen scenarios rather than reconstruction of training observations. A value of 1.0 indicates perfect directional alignment between predicted and observed margins; values above 0.7 indicate strong style correspondence. HSI_mean reports the per-route mean of HSI_t.

C. Multi-Scale Temporal Encoder

A lightweight causal temporal encoder maps the normalized window to a habit vector:

$$z_t = \mathrm{Enc}(X'_t) \in \mathbb{R}^d,$$

where $\mathrm{Enc}(\cdot)$ comprises three parallel TCN branches operating at different temporal scales (Fig. 2, right panel). We adopt three scales spanning immediate reactions to sustained patterns:

- **Short-horizon (1–2 s):** Captures instantaneous responses such as braking reaction and gap adjustments, matching typical driver reaction latencies.
- **Medium-horizon (5–10 s):** Encodes tactical patterns including preferred following distance, merge tempo, and intersection approach behavior.
- **Long-horizon (20–40 s):** Summarizes sustained tendencies—overall aggressiveness and comfort with traffic density—consistent with driver identification literature [1].

This multi-scale design reflects how driving habits manifest at different temporal granularities. Each TCN branch uses dilated causal convolutions with residual connections (Fig. 2, left panel). Branch outputs are concatenated, attention-pooled, and refined by a Head MLP to produce z_t. An exponential moving average stabilizes the representation online:

$$\bar{z}_t = \eta \bar{z}_{t-1} + (1-\eta) z_t, \quad \eta \in [0, 1).$$

The prediction head enforces nonnegativity via a smooth constraint:

$$\hat{e}_t = \mathrm{softplus}(W \bar{z}_t + U c_t + b) \in \mathbb{R}^4_{\geq 0}.$$

Training minimizes a weighted per-side regression with masking:

$$\mathcal{L}_t = \sum_{s \in \{f,b,l,r\}} w_s m_t^s \| \hat{e}_t^s - e_t^{\mathrm{act},s} \|_2^2,$$

where w_s balances side importance (front $>$ side $>$ back), prioritizing the most safety-critical direction.

D. CRI-Aware Personalized Envelope Estimation

Given habit vector $\bar{z}_t$, context c_t, and a directional CRI vector r_t, the decoder outputs a four-side envelope $e_t = [e_t^f, e_t^b, e_t^l, e_t^r] \in \mathbb{R}^4_{\geq 0}$ in meters. As introduced in Section III-A, CRI [4] supplies directional and global risk scalars derived from relative motion and lane/right-of-way context, normalized to $[0, 1]$ and filtered by RSS so larger values reflect stronger local hazard.

The decoder uses three specialized MLP heads (Fig. 2, right panel), each serving a distinct role:

- **Base head:** Outputs driver-specific baseline margins from $\bar{z}_t$, representing preferred spacing under nominal conditions.
- **Modulation head:** Scales the baseline using CRI, expanding margins on threatened sides proportionally to detected risk.
- **Reactive head:** Provides bounded corrections $\delta_t \in [-\Delta, \Delta]^4$ via $\tanh$, responding to transient situations.

This decomposition separates *driver preference* (Base) from *situational requirement* (Modulation) and *momentary adjustment* (Reactive), improving interpretability and enabling component-wise inspection. The final envelope combines these terms:

$$e_t = \mathrm{clip}_{\min}\Big(\underbrace{e_t^{\mathrm{base}}(\bar{z}_t) \odot m_t}_{\text{style scale}} + \underbrace{\pi_t(r_t, c_t)}_{\text{CRI prior}} + \underbrace{\delta_t(\bar{z}_t, r_t, c_t)}_{\text{reactive}} \Big), \tag{2}$$

where $m_t = \mathrm{softplus}(W_m \bar{z}_t + b_m) \geq 0$ rescales the base by habit, and $\pi_t = k_{\mathrm{cri}}(1 + k_{\mathrm{spd}} s(v_t)) r_t^{\{f,b,l,r\}}$ expands threatened sides with a speed-aware factor. Temporal smoothing via EMA ensures stable control integration:

$$\tilde{e}_t = \alpha \tilde{e}_{t-1} + (1-\alpha) e_t, \quad \alpha \in [0, 1). \tag{3}$$

a) Calibrated lower bounds.: To maintain formal safety guarantees and avoid pathological shrinkage, $\mathrm{clip}_{\min}$ enforces per-side minima: *front*—RSS longitudinal safe distance; *back*—1.0 m buffer; *left/right*—$\max\{0.25 \times \text{lane\_width}, 0.8\,\mathrm{m}\}$. These bounds ensure the personalized envelope never shrinks below RSS-consistent clearances.

E. Envelope Decoding to Control Adaptation

From the smoothed envelope $\tilde{e}_t$, we compute per-side intrusion depths and time-to-collision (TTC)-based severities to form a scalar threat level $\ell_t \in [0, 1]$ and a dominant direction θ_t^*. A simple aggregation,

$$\ell_t = \sigma\big(w_{\mathrm{TTC}} \phi_{\mathrm{TTC}}(t) + w_{\mathrm{pen}} \phi_{\mathrm{pen}}(t) + w_{\mathrm{sq}} \phi_{\mathrm{sq}}(t)\big), \tag{4}$$

combines (i) normalized minimum TTC over intruding agents, (ii) normalized penetration depth relative to $\tilde{e}_t$, and (iii) a "squeeze" cue measuring asymmetric side pressures; $\sigma(\cdot)$ is logistic and $w_\cdot$ are calibrated weights with front $>$ side $>$ back priority.

Let $u_0 = [a_0, \delta_0]$ be the base planner's acceleration/steer command and $u_s = [a_s, \delta_s]$ be a safety-adjusted fallback. We blend commands with a threat-dependent gain:

$$u_t = (1 - \lambda_t) u_0 + \lambda_t u_s, \quad \lambda_t = \min(1, k_\lambda \ell_t), \tag{5}$$

and gate emergency braking if $\min \mathrm{TTC} < \tau_{\mathrm{ebr}}$. Directional steering bias follows the envelope geometry, $\delta_s = \delta_0 + \kappa_{\mathrm{lat}}(e_t^r - e_t^l)$, which nudges away from the tighter side when forward progress is otherwise safe. This adaptation preserves the base planner's intent under low risk and escalates assistance as envelope violations or TTC hazards increase, providing a compact bridge from personalized, CRI-grounded representation to real-time actuation.

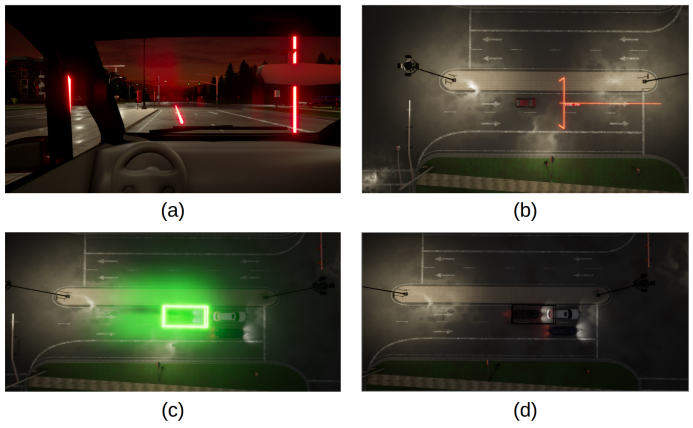

Fig. 3: Data collection environment and personalized-envelope visualization in CARLA. Data were collected from human drivers operating the ego vehicle via a Logitech G920 steering wheel, and subsequently de-identified prior to model training. (a) Driver's first-person viewpoint—red gates indicate the target route to follow during data collection. (b) Overhead monitoring view showing the ego vehicle (red) approaching a route gate; the orange marker indicates the next waypoint target. (c) Personalized safety envelope rendered in green when no vehicle is within the envelope boundary; the overlay is shifted upward by 4 m to avoid occluding on-board sensors. (d) Intrusion case—the envelope turns black when another vehicle enters the personalized safety margin, triggering the control adaptation layer.

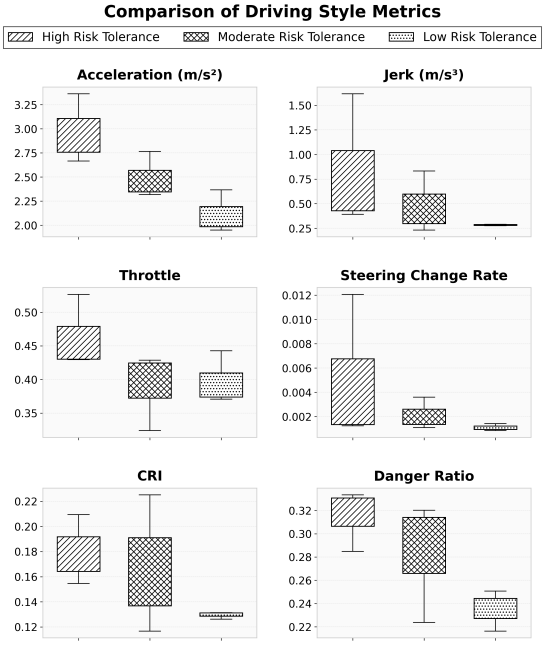

Fig. 4: Quantitative characterization of the three driver profiles used for PEDRA training, separated by risk-tolerance regime. Box plots show median (red line) and interquartile range across six behavioral signals. The High-RT profile exhibits markedly higher values on all axes: acceleration (+40%), jerk (+150%), and steering change rate ($4\times$) relative to Low-RT. CRI and Danger Ratio (fraction of frames with CRI > 0.25) confirm elevated risk exposure: the High-RT driver spends 33% of driving time in high-risk states versus 24% for Low-RT. The Moderate-RT profile falls consistently between the two extremes. These separations validate that the three datasets represent statistically distinct behavioral profiles rather than noisy samples of the same style.

IV. Experiments

We evaluate PEDRA by integrating it with Transfuser++ [21] in CARLA [22] on the Bench2Drive benchmark [5]. Our experiments address four questions: (1) Does personalization improve safety and comfort metrics across all routes? (2) Do improvements scale with personalization data? (3) Does PEDRA faithfully reproduce distinct risk-tolerance regimes, and how does profile choice affect safety? (4) What is the runtime overhead?

All experiments use CARLA's official Leaderboard harness in synchronous mode under the `SENSORS` track, with rendering and weather seeds fixed across all runs and identical sensor configurations for all methods. Bench2Drive provides 220 routes concentrated in accident-prone scenarios spanning dense urban intersections, high-speed merges, and complex multi-agent interactions; all 220 routes are executed per method with controller gains held constant and no online learning during testing. A high-risk subset of 44 routes is reserved for safety-critical evaluation: these routes were identified by running Transfuser++ across multiple independent evaluation runs and retaining only those on which collisions occurred consistently.

This study uses previously collected, de-identified human driver simulator data; the university's IRB determined this activity does not constitute direct human subjects research. Data were collected from human drivers operating the ego vehicle via a Logitech G920 steering wheel in CARLA on the Leaderboard `longest6` routes subset at 10 Hz, capturing ego kinematics, control inputs, lane context, and surrounding vehicle states. All personally identifiable information was removed prior to analysis and model training. From the resulting repository of de-identified logs, we selected three datasets maximally separated across six behavioral signals—acceleration, jerk, throttle, steering change rate, CRI, and danger ratio—yielding three risk-tolerance regimes: **High-RT** (high acceleration, short headways, elevated CRI exposure), **Low-RT** (low jerk, large safety margins, early intervention), and **Moderate-RT** (intermediate across all signals). Fig. 4 quantifies the behavioral separation across the three profiles. For the Moderate-RT profile, we construct checkpoints at 5 min, 10 min, and 15 min to study data efficiency; High-RT and Low-RT profiles use the full 15 min checkpoint. All profiles are frozen after training and evaluated without online adaptation.

The multi-scale encoder uses TCN branches with kernel sizes 3, 5, and 7 and dilation factors yielding receptive fields of approximately 2 s, 10 s, and 40 s at 10 Hz. The CRI module [4] computes an 8-sector directional risk summary

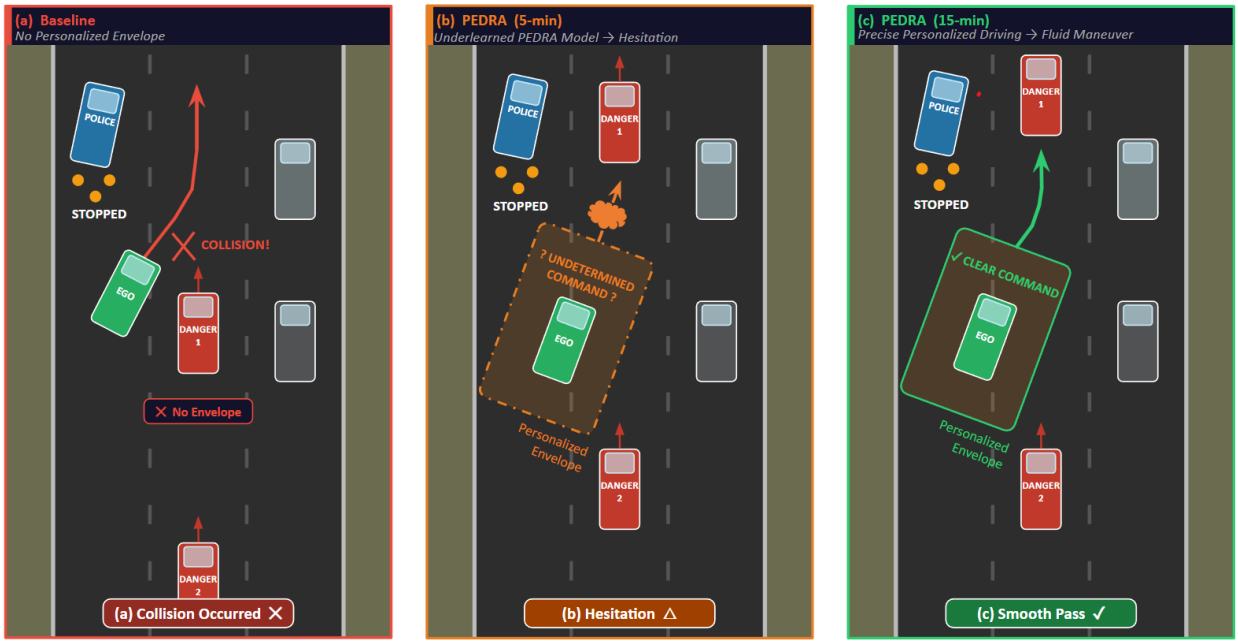

Fig. 5: Qualitative case study: the ego vehicle passes a stationary police car while two approaching vehicles (DANGER 1, DANGER 2) create a constrained merging window. **(a) Transfuser++**: without a personalized envelope, the planner steers into DANGER 1, causing a collision. **(b) Transfuser++ w/ PEDRA (5-min)**: the envelope detects DANGER 1 as an intrusion and holds position, but the undertrained habit representation hesitates after DANGER 1 clears, yielding a delayed and uncertain pass. **(c) Transfuser++ w/ PEDRA (15-min)**: the learned envelope (green) matches the driver's spacing habits, identifies the safe gap, and executes a smooth, decisive lane change.

plus a global score, collapsed to four cardinal directions for envelope decoding. EMA smoothing uses $\eta = 0.9$ for the habit vector and $\alpha = 0.8$ for the envelope. Training converges in approximately 50 epochs on 15 min of data; weights are frozen after convergence and stored as the driver profile.

We compare against two baselines. **Transfuser++** without any personalization module receives raw sensor inputs and produces control commands directly, isolating the contribution of the personalized envelope. **Transfuser++ w/ Driver2Vec** applies the Driver2Vec [1] framework to encode driver behavior as a 16-dimensional latent representation from the same driving logs, adapting the throttle and brake gains of the Transfuser++ controller at inference time; this baseline represents latent-embedding personalization methods that lack explicit geometric control semantics and formal safety bounds.

V. Evaluation

A. Qualitative Case Study

Fig. 5 examines a representative high-risk scenario in which the ego vehicle must pass a stationary police car while two approaching vehicles create a constrained merging window. This scenario directly tests the envelope's core function: detecting nearby threats, holding position when the gap is unsafe, and committing decisively once the gap clears.

Transfuser++, lacking any geometric safety reference, steers directly into the path of DANGER 1 and collides. Transfuser++ w/ PEDRA trained on 5 minutes of data correctly identifies the intrusion and holds position, but the underlearned habit vector cannot accurately characterize the driver's gap-acceptance threshold—producing hesitation and a delayed, uncertain pass. With 15 minutes of training data, the envelope precisely reflects the driver's spacing habits: the system immediately recognizes the clear gap after DANGER 1 passes and executes a smooth, decisive lane change with no unnecessary delay.

This progression—collision, hesitation, fluid maneuver—illustrates a key property of the geometric representation: envelope quality scales with training data, and that quality directly translates to behavioral confidence. We next validate this observation at scale.

B. Quantitative Analysis

Table I presents results across three panels. We report the following metrics throughout:

- **RSS/km**: Distance-normalized RSS violation intensity; lower is safer.
- **RSS viol rate**: Fraction of frames violating RSS bounds.
- **Jerk RMS**: Root-mean-square longitudinal jerk ($\mathrm{m\,s^{-3}}$); lower indicates smoother control.
- **HSI mean**: Habit Similarity Index (Eq. 1) measuring directional alignment between predicted envelope and observed spacing on held-out test routes; higher indicates greater style fidelity.

1) Data Efficiency: Moderate-RT Profile on All Routes: Panel 1 evaluates the Moderate-RT profile across increasing

TABLE I: Safety and comfort metrics across all experiments. Panel 1 demonstrates that PEDRA monotonically improves almost all metrics as training data grows, with the largest gains appearing between T++ and the 5 min checkpoint. Panel 2 shows that across all 220 routes, Low-RT achieves the strongest safety profile while High-RT preserves assertive behavior, confirming that PEDRA encodes individual style rather than collapsing to a single safe policy. Panel 3 isolates performance on structurally challenging scenarios: Low-RT reduces collisions and RSS violations most aggressively, while Moderate-RT achieves the highest habit fidelity (HSI mean 0.873). Arrows indicate preferred direction; bold denotes best per column. T++ = Transfuser++; D2V = Driver2Vec.

Panel 1 — T++ w/ PEDRA-Moderate-RT, All 220 Routes (data efficiency)

Metric	Dir.	T++	+PEDRA 5 min	+PEDRA 10 min	+PEDRA 15 min
RSS/km	↓	760.7	459.9	**450.0**	451.2
RSS viol rate	↓	0.2293	0.1336	0.1257	**0.1190**
Jerk RMS	↓	65.29	61.53	60.53	**58.94**
HSI mean	↑	0.723	0.787	0.846	**0.873**

Panel 2 — Cross-Method, All 220 Routes (all at 15 min)

Metric	Dir.	T++	T++ w/ D2V	T++ w/ PEDRA High-RT	T++ w/ PEDRA Moderate-RT	T++ w/ PEDRA Low-RT
Collisions/Route	↓	0.38	0.35	0.32	0.27	**0.26**
RSS viol rate	↓	0.2293	0.147	0.155	0.119	**0.031**
Jerk RMS	↓	65.29	63.0	58.94	46.8	**29.0**

Panel 3 — Cross-Method, 44 High-Risk Routes (all at 15 min)

Metric	Dir.	T++ w/ D2V	T++ w/ PEDRA High-RT	T++ w/ PEDRA Moderate-RT	T++ w/ PEDRA Low-RT
Collisions/Route	↓	0.977	0.862	0.682	**0.653**
Collision Rate (%)	↓	81.4	85.3	56.8	**53.0**
RSS viol rate	↓	0.285	0.303	0.160	**0.073**
Jerk RMS	↓	93.5	66.5	66.0	**40.8**
HSI mean	↑	0.747	0.756	**0.873**	0.807

data budgets on all 220 routes. Personalization yields consistent improvements in both safety-leading indicators and comfort metrics as data increases. Compared to Transfuser++, all personalized checkpoints reduce RSS/km by approximately 40% (760.7 $\rightarrow$ 450–460) and RSS violation rate by over 48% (0.229 $\rightarrow$ 0.119–0.134), indicating that the learned envelope enables earlier braking commitments and steadier gap-keeping—consistent with the hesitation effect observed in the case study at 5 minutes.

Comfort improves in parallel: Jerk RMS declines monotonically from 65.3 to 58.9 as data increases, consistent with a shift from late high-jerk braking to earlier low-slope deceleration. HSI mean rises from 0.723 to 0.873, confirming that longer observation windows yield envelopes that better match the driver's preferred margins. The gap between 10 min and 15 min is small (<1% on RSS/km), suggesting diminishing returns beyond 10 minutes; 15 min achieves the best comfort and fidelity and is preferable when data collection cost is not a constraint.

2) Cross-Method Comparison on All 220 Routes: Panel 2 compares all methods at 15 min on the full 220-route set. Transfuser++ w/ PEDRA-Low-RT achieves the strongest safety profile across the board: collisions per route drop from 0.38 (T++) to 0.26, and RSS violation rate falls by 86% (0.2293 $\rightarrow$ 0.031). Transfuser++ w/ D2V provides only marginal improvement over T++ (0.35 collisions/route), confirming that latent-embedding adaptation without geometric safety bounds offers limited benefit in accident-prone scenarios. Jerk RMS reflects the risk-tolerance spectrum: Low-RT yields the smoothest control (29.0), Moderate-RT is intermediate (46.8), and High-RT (58.94) approaches the unmodified T++ baseline (65.29)—preserving the assertive character of the source profile.

3) Style Fidelity and Safety: 44 High-Risk Routes: Panel 3 isolates performance on the 44 high-risk routes, revealing two complementary findings: PEDRA consistently outperforms Transfuser++ w/ D2V in safety metrics, and the learned envelope faithfully reproduces the source driver's risk-tolerance regime rather than converging to a single safe policy.

a) Safety advantage over D2V.: Transfuser++ w/ PEDRA-Low-RT achieves the strongest safety profile, with a collision rate of 53.0% and an RSS violation rate of 0.073—a 35% and 74% improvement over Transfuser++ w/ D2V (81.4% and 0.285) respectively. Transfuser++ w/ PEDRA-Moderate-RT also substantially outperforms D2V (56.8% collision rate, RSS violation rate 0.160), despite being trained on a less distinctive driving profile. Transfuser++ w/ D2V, which adapts only throttle and brake gains without geometric safety bounds, fails to provide consistent early intervention in high-risk scenarios.

b) Style fidelity.: The High-RT profile demonstrates that PEDRA does not suppress individual behavior in favor of safety: Transfuser++ w/ PEDRA-High-RT reproduces assertive

TABLE II: Runtime overhead on a high-density highway segment.

Overall	
Decision cycle mean (ms)	100.669
Personalized overhead mean (ms)	6.393
Overhead ratio	6.35%
Ticks (frames)	325
Per-component mean (ms)	
Envelope decoding	4.297
Intrusion checks	0.648
Control adaptation	1.346
Visualization / logging	0.102

driving patterns, reflected in its collision rate of 85.3% and RSS violation rate of 0.303—both exceeding even Transfuser++ w/ D2V. This is the expected consequence of faithfully encoding a risk-tolerance regime that tolerates shorter margins. Jerk RMS confirms the pattern: Low-RT yields the smoothest control (40.8), while High-RT (66.5) and Moderate-RT (66.0) both exhibit notably higher jerk—reflecting the more reactive braking characteristic of those profiles. HSI mean follows the fidelity ordering, with Moderate-RT achieving the highest score (0.873) given its longer training history across multiple data budgets, while Low-RT (0.807) and High-RT (0.756) reflect the trade-off between style expressiveness and safety constraint enforcement.

C. Runtime Overhead

Table II profiles runtime on a high-density highway segment (325 frames). The personalization module adds 6.4 ms per frame on average, with envelope decoding accounting for 67% of this overhead. The total decision cycle averages 100.7 ms, corresponding to an effective planning frequency of approximately 9.9 Hz.

VI. Conclusion and Future Work

We introduced PEDRA, which learns a personalized four-direction safety envelope from short driving sessions and deploys its margins as RSS-consistent planning constraints. On 220 Bench2Drive routes, PEDRA improves safety and comfort as personalization data increases while preserving distinct risk-tolerance profiles, with 6.4 ms overhead at 9.9 Hz. Current evaluation is simulation-only; future work will examine real vehicles, broader driver cohorts, adaptive RSS floors, and online habit adaptation.

Acknowledgement

This work is in part supported by the US National Science Foundation under Award No. NSF #2416937.

References

[1] J. Yang, R. Zhao, M. Zhu, D. Hallac, J. Sodnik, and J. Leskovec, "Driver2vec: Driver Identification from Automotive Data," *San Diego*, 2020.

[2] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2641–2646.

[3] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[4] B. Tian and W. Shi, "Context-aware risk assessment and its application in autonomous driving," in *Proceedings of the 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*. Gold Coast, Australia: IEEE, November 2025, p. TBD.

[5] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan, "Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving," *Advances in Neural Information Processing Systems*, vol. 37, pp. 819–844, 2024.

[6] S. Moosavi, P. D. Mahajan, S. Parthasarathy, C. Saunders-Chukwu, and R. Ramnath, "Driving Style Representation in Convolutional Recurrent Neural Network Model of Driver Identification," Feb. 2021, arXiv:2102.05843 [cs]. [Online]. Available: http://arxiv.org/abs/2102.05843

[7] W. Dong, T. Yuan, K. Yang, C. Li, and S. Zhang, "Autoencoder regularized network for driving style representation learning," *arXiv preprint arXiv:1701.01272*, 2017.

[8] L. Lu, "Learning driving style embedding from gps-derived moving patterns for driver identification," *arXiv preprint arXiv:2401.06986*, 2024.

[9] X. Yao, S. C. Calvert, and S. P. Hoogendoorn, "Driving heterogeneity identification using machine learning: A review and framework for analysis," *Transportation Research Interdisciplinary Perspectives*, vol. 32, p. 101511, 2025.

[10] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 128–14 147, 2022.

[11] Y. Zhou, M. Lu, X. Liu, Z. Che, Z. Xu, J. Tang, Y. Zhang, Y. Peng, and Y. Peng, "Distributional generative adversarial imitation learning with reproducing kernel generalization," *Neural Networks*, vol. 165, pp. 43–59, 2023.

[12] M. Yildirim and S. Fallah, "Human-like autonomous driving on dense traffic," *arXiv preprint arXiv:2310.02477*, 2023.

[13] X. Wen, X. Zheng, Z. Cui, S. Jian, and D. He, "Preference-based reinforcement learning for autonomous vehicle control considering the benefits of following vehicles," *IEEE Transactions on Intelligent Vehicles*, 2024.

[14] Y. Zhao, J. E. Escamill, W. Lu, and H. Wang, "Ra-pbrl: Provably efficient risk-aware preference-based reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 60 835–60 871, 2024.

[15] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[16] J. Bernhard, P. Hart, A. Sahu, C. Schöller, and M. G. Cancimance, "Risk-based safety envelopes for autonomous vehicles under perception uncertainty," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 104–111.

[17] Z. Yang, X. Jia, H. Li, and J. Yan, "Llm4drive: A survey of large language models for autonomous driving," *arXiv preprint arXiv:2311.01043*, 2023.

[18] C. Cui, Z. Yang, Y. Zhou, Y. Ma, J. Lu, L. Li, Y. Chen, J. Panchal, and Z. Wang, "Personalized Autonomous Driving with Large Language Models: Field Experiments," May 2024, arXiv:2312.09397 [cs]. [Online]. Available: http://arxiv.org/abs/2312.09397

[19] X. Han, X. Chen, Z. Cai, P. Cai, M. Zhu, and X. Chu, "From words to wheels: Automated style-customized policy generation for autonomous driving," *arXiv preprint arXiv:2409.11694*, 2024.

[20] G. Kou, F. Jia, W. Mao, Y. Liu, Y. Zhao, Z. Zhang, O. Yoshie, T. Wang, Y. Li, and X. Zhang, "Padriver: Towards personalized autonomous driving," *arXiv preprint arXiv:2505.05240*, 2025.

[21] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8240–8249.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.