Interpretable & Efficient Deep RL for Autonomous Driving

Mohammadamin Kiani, Danial Parnian

Department of Computer Engineering, Sharif University of Technology
{mohammadamin.kiani01, danial.parnian01}@sharif.edu

Interpretable & Efficient Deep RL for Autonomous Driving

Danial Parnian   Mohammadamin Kiani
Deep Reinforcement Learning, September 2025

Motivation & Problem

- End-to-end RL can adapt but tends to be black-box; safety/legal require **interpretability**.
- Two complementary angles:
  - (A) **Latent world-model + MaxEnt RL**: interpretable *perception* via decoded bird's-eye masks.
  - (B) **ICCT**: interpretable *control* via small crisp trees with sparse linear leaves.
- Goal: **trustworthy**, **robust** urban driving — fast learning, safe behavior, human-auditable decisions.

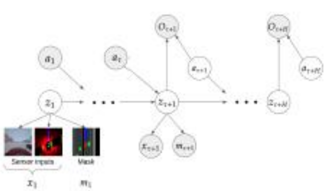
(A) Latent MaxEnt RL: Formulation

MDP: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, T, \gamma, \rho_0)$, policy $\pi(a|s)$.
MaxEnt RL in latent state z_t:

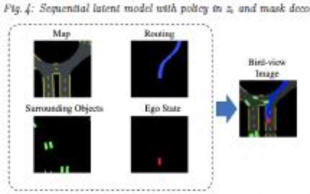
$$\max_{\phi} \mathbb{E}\left[\sum_{t=1}^{H} r(z_t, a_t) - \log \pi_\phi(a_t|z_t)\right]$$

(Optimized with SAC for stability/exploration.)
Mask quality (avg. pixel diff.): $e = \frac{1}{N} \sum_t \frac{\|\hat{m}_t - m_t\|_1}{W \times H \times C}$

Model & Decoder



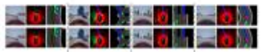
Chen Fig. 4: Sequential latent model with policy in z_t and mask decoder.



Chen Fig. 6: Bird's-eye semantic mask (map, route, objects, ego).

Key points
- Multi-modal inputs (camera+LiDAR) $\rightarrow$ compact z_t.
- Decode z_t to a $64\times64\times3$ mask to explain perception.
- Train jointly: variational sequential model + SAC on z_t; mask supervised only during training.
- Reward shaping: lane-keeping, speed compliance, collision/lat-accel penalties.

Results & Failure Modes



Chen Fig. 9: Reconstructions.

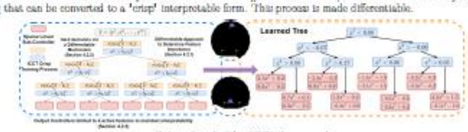
Chen Fig. 8: Learning curves.

- Latent-RL variants learn **faster** and reach **higher** asymptotes than classic deep RL baselines.
- Masks remain faithful (mean error ≈ 0.032), enabling human inspection.
- **Failures:** Rare/occluded objects can degrade masks, preceding control errors.

(B) ICCT: Interpretable Control via Differentiable Trees

Goal: Directly learn a policy $\pi_\theta(a|s)$ represented as a small, human-readable decision tree.
Interpretable Continuous Control Tree (ICCT): A tree where:
- **Decision nodes** are crisp rules on a *single* state feature: $x_k > b_i$.
- **Leaf nodes** are sparse linear controllers: $a_d = \Sigma_j \beta_{dj} x_j + \delta_d$

Key Idea: Differentiable Crispification. To enable gradient-based RL, the model uses a 'fuzzy' form during training that can be converted to a 'crisp' interpretable form. This process is made differentiable.



Paleja Fig. 1: The ICCT framework.

Differentiable Tree-Building:
1. **Node Crispification:** A differentiable 'one-hot' function selects the single most important feature for the decision rule.
2. **Outcome Crispification:** A second 'one-hot' function converts the sigmoid probability into a hard left/right branch decision.
3. **Sparse Leaf Controller:** A 'k-hot' selection identifies the most salient features for the linear controller at each leaf.

This allows direct optimization of a transparent policy using standard RL algorithms like SAC.

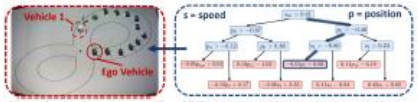
Algorithm 1: ICCT Action Choice

Input: ICCT $\mathcal{I}(\cdot)$, state $\mathbf{x}$, sparsity e, training flag t
Output: action $\mathbf{a}$

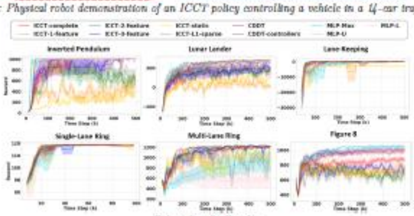
NODE_CRISP: $\sigma(\alpha(\mathbf{w}^T\mathbf{x} - b)) \rightarrow \sigma(\alpha(w_k x_k - b))$
OUTCOME_CRISP: $\sigma(\dots) \rightarrow \mathbb{1}(\alpha(w_k x_k - b) > 0)$
$l_d \leftarrow$ INTERPRETABLE_NODE_ROUTING($\mathbf{x}$)
$l_d' \leftarrow$ ENFORCE_CONTROLLER_SPARSITY(e, l_d)
if training flag t is TRUE **then**
  Sample $\mathbf{a} \sim \mathcal{N}(l_d'(\mathbf{x}), \gamma_d)$ (exploration)
else
  $\mathbf{a} \leftarrow l_d'(\mathbf{x})$ (exploitation)
end if

ICCT Results

ICCTs produce policies that are not only interpretable but also high-performing and efficient.



Paleja Fig. 5: Physical robot demonstration of an ICCT policy controlling a vehicle in a 14-car traffic scenario.



Paleja Fig. 8: Results.

Quantitative Highlights:
- **High Performance:** Matches or **outperforms** deep black-box models (MLPs) by up to 33% in complex autonomous driving scenarios.
- **Extreme Parameter Efficiency:** Achieves top performance with a **300x-600x reduction** in the number of policy parameters compared to deep learning baselines.
- **Verifiable & Robust:** The simple tree structure is amenable to formal verification and was demonstrated on a 14-car physical robot platform, proving real-world applicability.

Methodology Comparison:

Both papers target interpretability in AD, but focus on different parts of the problem.
Paper (A) - Latent MaxEnt RL:
- **Focus: Interpretable Perception.**
- **Answers:** *"What does the agent see?"*
- **Method:** Learns a compressed latent state z_t and uses a decoder to translate it into a human-understandable bird's-eye view mask.
- **Limitation:** The control policy $\pi(a|z_t)$ is still a black-box MLP.

Paper (B) - ICCT:
- **Focus: Interpretable Control.**
- **Answers:** *"Why did the agent take this action?"*
- **Method:** The policy itself is a white-box decision tree. The path from state to action is explicit and traceable.
- **Limitation:** Assumes a pre-processed, meaningful state vector.

Synergy: The two approaches are highly complementary. One could build a fully interpretable system by using model (A) to generate semantic features from raw sensor data, which are then fed into the transparent ICCT policy (B).

References

[1] Chen, Li, Tomizuka. *Interpretable End-to-End Urban Autonomous Driving with Latent Deep RL.* arXiv:2001.08726 (2020).
[2] Paleja, Niu, Silva, et al. *Learning Interpretable, High-Performing Policies for Autonomous Driving.* arXiv:2202.02352 (2023).
[3] Prakash, Avi, et al. *Efficient and Generalized end-to-end Autonomous Driving System with Latent Deep Reinforcement Learning and Demonstrations.* arXiv:2206.15806 (2022).

LaTeX TikZposter

Motivation & Problem

- End-to-end RL can adapt but tends to be black-box; safety/legal require **interpretability**.
- Two complementary angles:
  - (A) **Latent world-model + MaxEnt RL**: interpretable *perception* via decoded bird's-eye masks.
  - (B) **ICCT**: interpretable *control* via small crisp trees with sparse linear leaves.
- Goal: **trustworthy**, **robust** urban driving — fast learning, safe behavior, human-auditable decisions.

(A) Latent MaxEnt RL: Formulation

MDP: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, T, \gamma, \rho_0 \rangle$, policy $\pi(a \,|\, s)$.

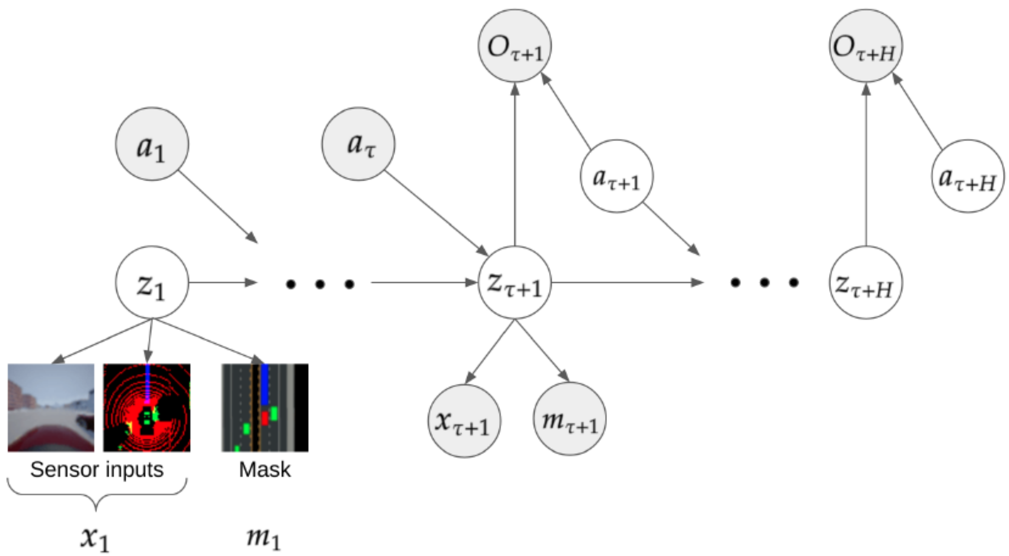
MaxEnt RL in latent state z_t:

$$\max_{\phi} \; \mathbb{E}\Big[\sum_{t=1}^{H} r(z_t, a_t) - \log \pi_{\phi}(a_t \,|\, z_t)\Big]$$

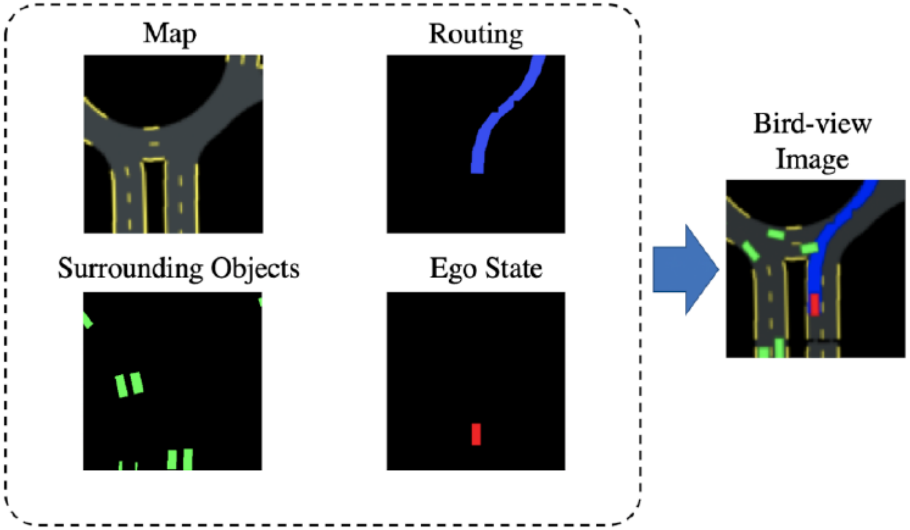
(Optimized with SAC for stability/exploration.)

Mask quality (avg. pixel diff.): $e = \frac{1}{N} \Sigma_i \frac{\|\hat{m}_i - m_i\|_1}{W \times H \times C}$

Model & Decoder



Chen Fig. 4: Sequential latent model with policy in z_t and mask decoder.

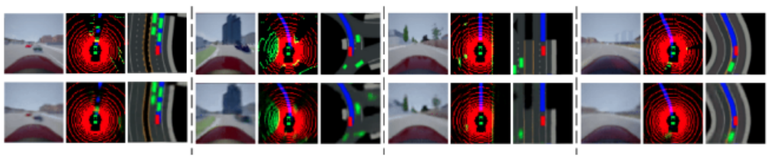


Chen Fig. 6: Bird's-eye semantic mask (map, route, objects, ego).

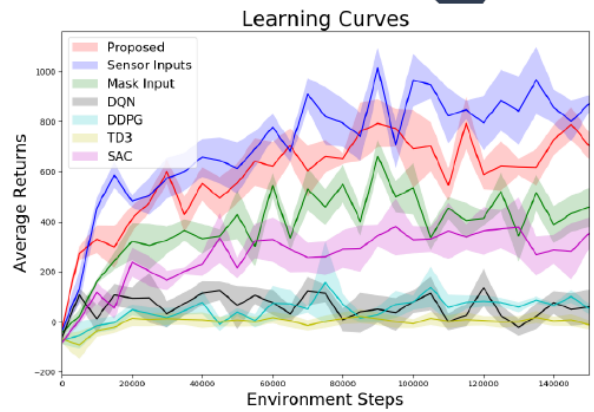
Key points

- Multi-modal inputs (camera+LiDAR) $\rightarrow$ compact z_t.
- Decode z_t to a $64\times64\times3$ mask to *explain* perception.
- Train jointly: variational sequential model + SAC on z_t; mask supervised only during training.
- Reward shaping: lane-keeping, speed compliance, collision/lat-accel penalties.

Results & Failure Modes



Chen Fig. 9: Reconstructions.



Chen Fig. 8: Learning curves.

- Latent-RL variants learn **faster** and reach **higher** asymptotes than classic deep RL baselines.
- Masks remain faithful (mean error ≈ 0.032), enabling human inspection.
- **Failures:** Rare/occluded objects can degrade masks, preceding control errors.

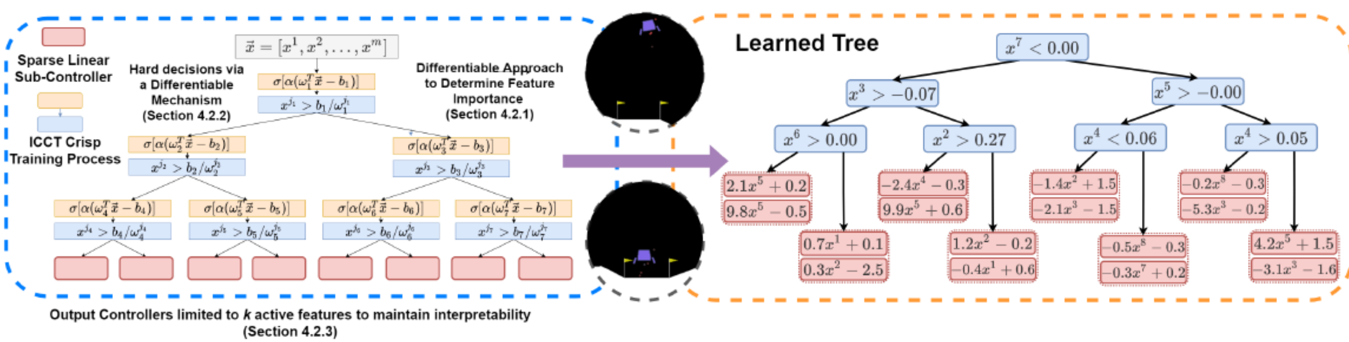
(B) ICCT: Interpretable Control via Differentiable Trees

Goal: Directly learn a policy $\pi_\theta(a|s)$ represented as a small, human-readable decision tree.

Interpretable Continuous Control Tree (ICCT): A tree where:

- **Decision nodes** are crisp rules on a *single* state feature: $x_k > b_i$.
- **Leaf nodes** are sparse linear controllers: $a_d = \Sigma \beta_{dj} x_j + \delta_d$.

Key Idea: Differentiable Crispification. To enable gradient-based RL, the model uses a "fuzzy" form during training that can be converted to a "crisp" interpretable form. This process is made differentiable.



Paleja Fig. 1: The ICCT framework.

Differentiable Tree-Building:

1. **Node Crispification:** A differentiable ‘one-hot‘ function selects the single most important feature for the decision rule.
2. **Outcome Crispification:** A second ‘one-hot‘ function converts the sigmoid probability into a hard left/right branch decision.
3. **Sparse Leaf Controller:** A ‘k-hot‘ selection identifies the most salient features for the linear controller at each leaf.

This allows direct optimization of a transparent policy using standard RL algorithms like SAC.

Algorithm 1: ICCT Action Choice

Input: ICCT $\mathcal{I}(\cdot)$, state $\mathbf{x}$, sparsity e, training flag t
Output: action $\mathbf{a}$

1: NODE_CRISP: $\sigma(\alpha(\tilde{\mathbf{w}}^T\mathbf{x} - b)) \rightarrow \sigma(\alpha(w_k x_k - b))$

2: OUTCOME_CRISP: $\sigma(\dots) \rightarrow \mathbf{1}(\alpha(w_k x_k - b) > 0)$

3: $l_d \leftarrow$ INTERPRETABLE_NODE_ROUTING($\mathbf{x}$)

4: $l'_d \leftarrow$ ENFORCE_CONTROLLER_SPARSITY(e, l_d)

5: **if** training flag t is TRUE **then**

6: Sample $\mathbf{a} \sim \mathcal{N}(l'_d(\mathbf{x}), \gamma_d)$ (exploration)

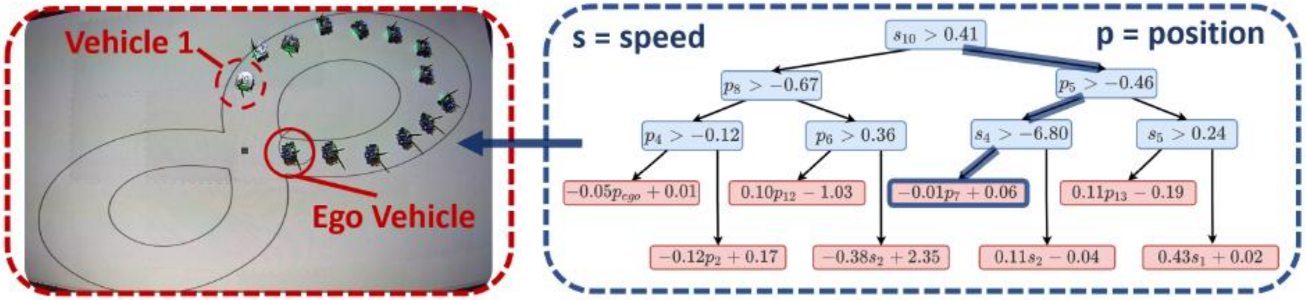
7: **else**

8: $\mathbf{a} \leftarrow l'_d(\mathbf{x})$ (exploitation)

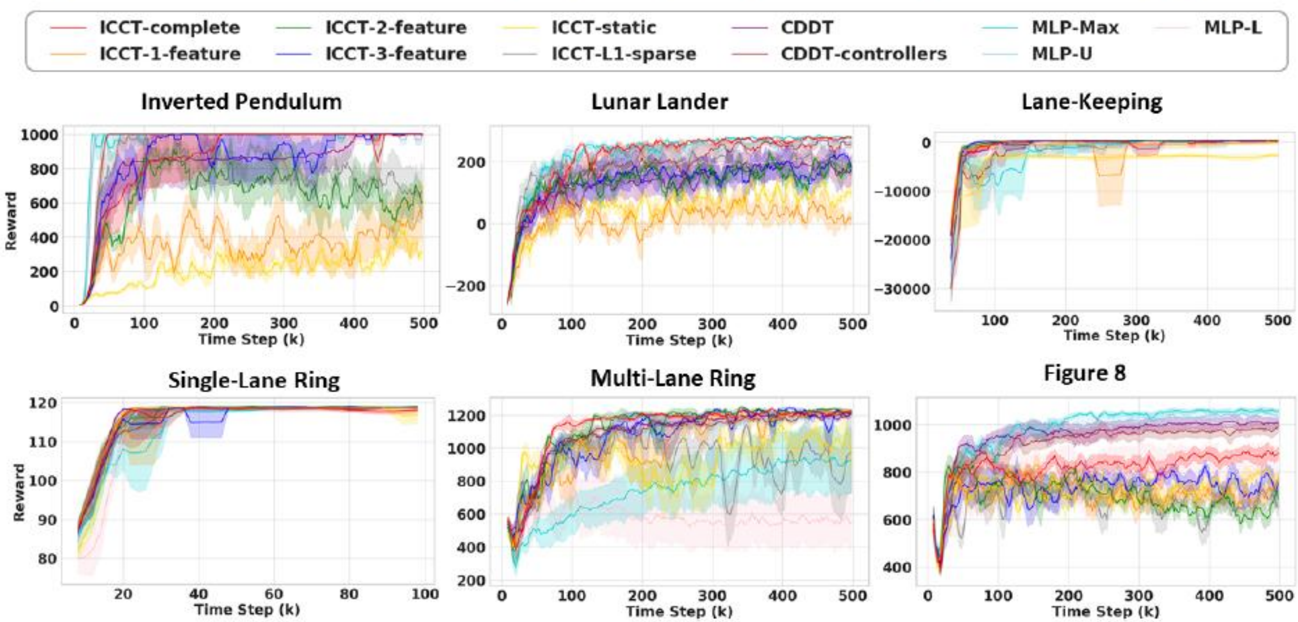
9: **end if**

ICCT Results

ICCTs produce policies that are not only interpretable but also high-performing and efficient.



Paleja Fig. 5: Physical robot demonstration of an ICCT policy controlling a vehicle in a 14-car traffic scenario.



Paleja Fig. 8: Results.

Quantitative Highlights:

- **High Performance:** Matches or **outperforms** deep black-box models (MLPs) by up to 33% in complex autonomous driving scenarios.
- **Extreme Parameter Efficiency:** Achieves top performance with a **300x-600x reduction** in the number of policy parameters compared to deep learning baselines.
- **Verifiable & Robust:** The simple tree structure is amenable to formal verification and was demonstrated on a 14-car physical robot platform, proving real-world applicability.

Methodology Comparison:

Both papers target interpretability in AD, but focus on different parts of the problem.

Paper (A) - Latent MaxEnt RL:

- **Focus: Interpretable Perception**.
- **Answers:** *"What does the agent see?"*
- **Method:** Learns a compressed latent state z_t and uses a decoder to translate it into a human-understandable bird's-eye view mask.
- **Limitation:** The control policy $\pi(a|z_t)$ is still a black-box MLP.

Paper (B) - ICCT:

- **Focus: Interpretable Control**.
- **Answers:** *"Why did the agent take this action?"*
- **Method:** The policy itself is a white-box decision tree. The path from state to action is explicit and traceable.
- **Limitation:** Assumes a pre-processed, meaningful state vector.

Synergy: The two approaches are highly complementary. One could build a fully interpretable system by using model (A) to generate semantic features from raw sensor data, which are then fed into the transparent ICCT policy (B).

References

[1] Chen, Li, Tomizuka. *Interpretable End-to-End Urban Autonomous Driving with Latent Deep RL.* arXiv:2001.08726 (2020).
[2] Paleja, Niu, Silva, et al. *Learning Interpretable, High-Performing Policies for Autonomous Driving.* arXiv:2202.02352 (2023).
[3] Prakash, Avi, et al. *Efficient and Generalized end-to-end Autonomous Driving System with Latent Deep Reinforcement Learning and Demonstrations.* arXiv:2205.15805 (2022).