Sim2Real: Improving Autonomous Driving Performance by Learning from World Models with Privileged Information

Team Members: Jack Liu, Ryan Catullo, Mac Ya, Sunny Yu

Emails: jiayiliu@stanford.edu, rcatullo@stanford.edu, maaac@stanford.edu, syu03@stanford.edu

Extended Abstract Autonomous driving (AD) systems trained in simulation using world model-based reinforcement learning (RL) have shown impressive task performance, but often rely on privileged information unavailable in real-world settings. This limits their deployability and raises the challenge of sim-to-real transfer. Our project proposes a student-teacher framework to address this issue by distilling structured latent knowledge from a simulation-trained expert model into a student trained only on observable inputs such as camera and LiDAR data.

We leverage the CarDreamer platform and use models trained with privileged data to align their latent representations with those of a student model using KL-divergence regularization. The student uses an identical DreamerV3 architecture but is trained on sensor-level inputs without access to ground-truth BEV or intent information. This alignment encourages the student to internalize high-level planning competence without relying on unrealistic input assumptions.

Our implementation involves modifying the CarDreamer pipeline, resolving environment compatibility and checkpoint issues, and training the student model using latent distillation losses. Additionally, for our baseline evaluations, we removed all privileged information and retrained the models. We evaluate our approach on three CARLA tasks of increasing difficulty: Left Turn Simple, Four Lane, and Navigation. Results show that the student-teacher model outperforms a baseline model trained without privileged information, achieving a 92.31% success rate on the Four Lane task (vs. 25.23% baseline) and demonstrating comparable behavior to the teacher in the Navigation task.

While promising, our approach has limitations. It assumes temporally aligned representation spaces and inherits the teacher's biases. Furthermore, simulation environments still fall short of capturing real-world unpredictability. Nonetheless, our framework offers a scalable path to real-world AD policy learning by exploiting simulation knowledge while relaxing deployment constraints. Future work includes extending to multi-agent settings and exploring alternative alignment objectives such as contrastive learning or domain adversarial training.

Our results demonstrate the potential of latent-space knowledge transfer as a mechanism for bridging the sim-to-real gap in autonomous systems, paving the way for robust sensor-only agents trained via efficient simulation-based curricula.

Abstract

This project aims to improve the real-world applicability of autonomous driving (AD) agents trained using world models and reinforcement learning in simulation. While recent SOTA world-model based RL methods like CarDreamer and Think2Drive demonstrate impressive performance in simulated environments like CARLA, they rely heavily on privileged information (i.e. information not available directly to cars in the real world, such as BEV image, traffic light information, other vehicles' intent, etc). These inputs are almost always unavailable to real-world autonomous systems, meaning that all of these models can only be used as "experts" in simulators, but are not deployable in a real system. To bridge this gap, our project aims to train with and without privileged information using KL-divergence to enforce latent-space alignment. Specifically, we train two world models — one with privileged information and one without — and use KL divergence to align their latent representations for the student model to learn from the teacher model without having direct access to privileged information during the training process. This enables the model trained without privileged inputs to learn from the richer world understanding of the privileged one. We evaluate the model on three tasks from the CarDreamer environment at three difficulty levels: the left turn easy task, the four lane task, and the navigation task. Our results show that our student-teacher model outperforms the baseline model (an agent trained without privileged information) and achieves a success rate of 92.3% on the four lane task which is significantly higher than the baseline performance. Additional analysis shows that the agents can effectively navigate the environment, showing the effectiveness of our approach.

1 Introduction

Recent advances in world model-based reinforcement learning (RL) have significantly improved the capabilities of autonomous driving (AD) agents in simulation environments. In particular, frameworks such as CarDreamer Gao et al. (2024) and Think2Drive Li et al. (2024) have demonstrated state-of-the-art performance on complex driving tasks by leveraging learned latent representations and high-capacity temporal dynamics models. A key feature shared by these systems is their use of *privileged information*—such as bird's-eye view (BEV) maps, ground-truth vehicle states, and traffic light positions—as inputs to the world model. These inputs are typically extracted directly from the simulator and enable precise planning and prediction, bypassing the need for raw sensor fusion or perception.

While this approach accelerates learning and stabilizes policy optimization, it raises a critical deployment issue: real-world autonomous systems do not have access to such privileged information at test time. Instead, they must rely on imperfect, noisy sensor data like RGB images, LiDAR, and IMU signals. Consequently, models trained in simulation with privileged information often fail to generalize to real-world conditions—a challenge broadly referred to as the sim-to-real gap.

Our project addresses this gap by introducing a student-teacher training paradigm for autonomous driving that enables policy learning without privileged inputs, while still benefiting from the structured world knowledge available in simulation. Specifically, we use the open-sourced checkpoints from CarDreamer (which trains their models with privileged information) as the teacher model to train the student model using only observable environmental data. To transfer high-level planning competencies from teacher to student, we align their latent state representations using KL-divergence regularization during training. This latent distillation enables the student to approximate the teacher's internal representation space without direct access to privileged information.

By integrating latent-space alignment into the world model learning process, our method preserves the strengths of high-performing simulation-based agents while producing a model suitable for deployment under real-world sensory constraints. We evaluate our method on three benchmark tasks in the CarDreamer environment—Left Turn (Easy), Four Lane, and Navigation—and report substantial improvements over baseline models trained without privileged information. Our results demonstrate that student models can inherit meaningful structure from privileged training even when direct inputs are unavailable, offering a promising direction for robust sim-to-real transfer in AD systems.

2 Related Work

2.1 Reinforcement Learning for AD

Deep reinforcement leanring is a critical method for training autonomous driving agents Kiran et al. (2021); Sallab et al. (2017). In fact, simulation-based reinforcement learning has been a common approach that trains then adapts autonomous driving agents from a simulated to real environment Pan et al. (2017); Osiński et al. (2020). Some most recent examples that use reinforcement learning to train AD agents include CarDreamer Gao et al. (2024) and Think2Drive Li et al. (2024), which both build on the DreamerV3 Hafner et al. (2024) and adapt it for autonomous driving.

Other existing RL methods for AD include Abeysirigoonawardena et al. (2019), which uses Bayesian optimization to synthesize adversarial scenarios in which self-driving agents are more likely to fail, and subsequently fine-tunes the policy using imitation learning. Feng et al. (2021) and Song et al. (2023) actively train adversarial agents or generate failure trajectories in simulation. Feng et al. (2021) formulates adversarial background vehicles as learning agents that perturb the environment to stress-test the ego vehicle's policy. Similarly, ACERO Song et al. (2023) formalizes a trajectory generation framework that causes mission-critical failures under predefined safety rules in CARLA. While different in the specific training objectives, these approaches have shown that reinforcement learning is an effective training paradigm for autonomous driving.

2.2 Student-Teacher Framework for AD

The teacher-student technique is not new for autonomous driving: Khosravian et al. (2022) is a recent study leverages a semi-supervised teacher-student framework for semantic segmentation in

autonomous driving, demonstrating that a student model trained on a different domain can closely match the performance of supervised models. Similarly, Li et al. (2022) is a multi-task imitation learning framework that enables automated coaching for complex motor skills like performance driving, using self-supervised signals from non-interactive data to overcome the scarcity of expert-student interactions, and demonstrating effectiveness through both simulation and real-world human-subject studies.

In contrast to these approaches, our work contributes a novel learning-based sim-to-real transfer strategy by explicitly aligning the latent state representations of a privileged-input world model (teacher) with those of a model trained only on observable data (student). While inspired by the teacher-student paradigms in distillation and imitation learning, our method integrates this alignment into the world model training itself via KL-regularized latent distillation. This enables the student model to inherit the planning competencies of privileged agents without relying on privileged inputs at inference time.

3 Methods

3.1 Data and Experimental Setup

We train and evaluate the performance of autonomous driving (AD) models in closed-loop environments using CARLA Dosovitskiy et al. (2017) as the simulation environment. The environment includes dynamic traffic participants and detailed infractions (collisions, traffic light violations, etc.) for evaluating autonomous driving. We build on CarDreamer Gao et al. (2024), an open-source learning platform designed for training and evaluating world model based autonomous driving algorithm that leverages dreamerv3 Hafner et al. (2023) to perform our training. We chose CarDreamer because it has the only open-sourced repository that leverages the dreamerv3 framework (which serves as a backbone for configuring RL algorithms for diverse tasks) for world-model based RL in CARLA.

As a sanity check, we first run the Car Dreamer checkpoints to ensure that the teacher model performs optimally on the task. We will then use the checkpoints as our teacher model. From results in Tab 4, we can see that the model has an optimal performance that matches the results reported in Gao et al. (2024). Therefore, we know that by aligning the latent space of the student model with the teacher model, we are aligning to a model with good performance on the benchmark.

3.2 Problem Formulation

Our focus is on restricting the input of privileged information x_t as in Li et al. (2024) to information only available in real-life driving scenarios, i.e. outside of a simulation. The action space A stays the same: throttle, steer, brake. We decompose our state formulation into *privileged* p_t and *environmental* o_t information, $x_t = (p_t, o_t)$. In CARLA, this means o_t consists of camera (270° of rotation), LiDAR, and collision sensors.

We give a brief overview of the DreamerV3 architecture, which is used to train the teacher models on privileged information.

3.3 World Model

The idea behind the world model is to learn compact representations of sensor information via autoencoding. Then a neural planner predicts future representations and rewards to predict actions. The world model is implemented as a **Recurrent State-Space Model (RSSM)**.

An encoder q_ϕ maps raw CARLA inputs x_t to latent representations z_t. A sequence model with a recurrent hidden state h_t predicts the dynamics of the latent world given actions a_t.

$$
\begin{aligned}
&\text{Sequence model:} && h_t = f_\phi(h_{t-1},\, z_{t-1},\, a_{t-1}),\\
&\text{Encoder:} && z_t \sim q_\phi(z_t \mid h_t,\, x_t),\\
&\text{Dynamics predictor:} && \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t),\\
&\text{Reward predictor:} && \hat{r}_t \sim p_\phi(\hat{r}_t \mid h_t,\, z_t),\\
&\text{Continue predictor:} && \hat{c}_t \sim p_\phi(\hat{c}_t \mid h_t,\, z_t),\\
&\text{Decoder:} && \hat{x}_t \sim p_\phi(\hat{x}_t \mid h_t,\, z_t)
\end{aligned}
\tag{1}
$$

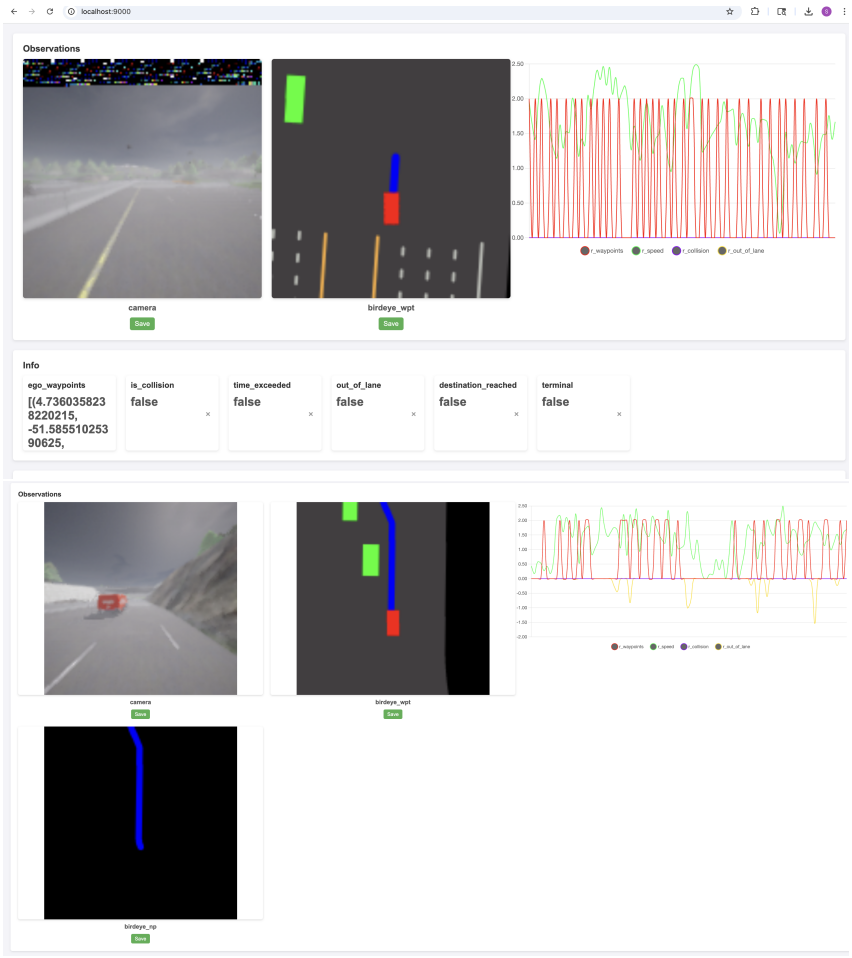

Figure 1: The visualization of the evaluation platform provided by CarDreamer for the CarDreamer checkpoint (top) and our Student-Teacher Model (bottom) on the task. The left panel shows the camera inputs, and the right panel shows the bird-eye view inputs. From the log, we can see that the collision rate and out-of-lane rate is almost always 0, while the speed oscillates around 2. Similarly, we can see that while the speed also oscillates, the collision rate remains 0 for our model, showing optimal performance.

The reward predictor predicts reward in the latent world model, and continue predicts the terminal state of the environment in the latent world model.

As in Gao et al. (2024), the encoder and decoder use CNNs for image inputs and MLPs for vector inputs. The reward, dynamics, and continue heads are naturally MLPs as well. The entire model is trained end-to-end, so actions a_{t-1} in the sequence model come from the neural planner described next.

The world model has three loss functions, $\mathcal{L}_{\text{dyn}}, \mathcal{L}_{\text{pred}}, \mathcal{L}_{\text{rep}}$ for dynamics, prediction, and representation. They have corresponding weight hyperparameters $\beta_{\text{dyn}} := 1, \beta_{\text{pred}} := 1, \beta_{\text{rep}} := 0.1$ as in DreamerV3:

$$\mathcal{L}(\phi) := \mathbb{E}_{q_\phi}\left[\sum_{t=1}^{T}\left(\beta_{\text{pred}}\mathcal{L}_{\text{pred}}(\phi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\phi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}(\phi)\right)\right].$$

The prediction loss steers the decoder and reward head via a symmetrized log-squared penalty and the "continue" head via a logistic-regression loss. The dynamics loss forces the recurrent state model to match its one-step latent forecast $p_\phi(z_t \mid h_t)$ to the encoder's posterior $q_\phi(z_t \mid h_t, x_t)$. The representation loss encourages the latent encoding to become more predictable, thus enabling a

factorized dynamics predictor for efficient "imagination." Both KL-based terms use a stop-gradient on one side (denoted $\mathrm{sg}(\cdot)$) and are clipped to a minimum of one nat (≈ 1.44 bits) via free-bits regularization, preventing the model from collapsing to trivial solutions. Formally:

$$\begin{aligned}
\mathcal{L}_{\mathrm{pred}}(\phi) &= -\ln p_\phi(x_t \mid z_t,\, h_t) \;-\; \ln p_\phi(r_t \mid z_t,\, h_t) \;-\; \ln p_\phi(c_t \mid z_t,\, h_t),\\
\mathcal{L}_{\mathrm{dyn}}(\phi) &= \max\Big(1,\; \mathrm{KL}\big[\,\mathrm{sg}\big(q_\phi(z_t \mid h_t, x_t)\big) \;\big\|\; p_\phi(z_t \mid h_t)\big]\Big),\\
\mathcal{L}_{\mathrm{rep}}(\phi) &= \max\Big(1,\; \mathrm{KL}\big[\,q_\phi(z_t \mid h_t, x_t) \;\big\|\; \mathrm{sg}\big(p_\phi(z_t \mid h_t)\big)\big]\Big).
\end{aligned}$$

3.4 Actor–Critic Learning

We train the actor and critic networks end-to-end on trajectories imagined by the latent world model. The actor and critic's states are given b concatenated tuples $s_t = \{h_t, z_t\}$ of the hidden state and encoded latent representation. The actor aims to maximize discounted future predicted returns by the world model with a time horizon of $T = 16$. The remaining rewards are estimated by the critic: $v_\phi(R_t \mid s_t)$.

The world model and actor generate a trajectory of imagined rollouts $s_{1:T}, a_{1:T}, r_{1:T}, c_{1:T}$. To estimate returns that consider rewards beyond the prediction horizon, DreamerV3 computes bootstrapped λ-returns that integrate the predicted rewards and the values. The critic v_ϕ is optimized by regressing to λ-returns computed from imagined rewards $\hat{r}_t$ and discount factor γ:

$$\mathcal{L}_{\mathrm{critic}}(\phi) := -\sum_{t=1}^{T} \ln p_\phi(R_t^\lambda \mid s_t), \quad R_t^\lambda := r_t + \gamma c_t((1-\lambda)v_t + \lambda R_{t+1}^\lambda), \quad R_T^\lambda := v_T \tag{2}$$

The actor learns to choose actions that maximize return while exploring through an entropy regularizer.

Dreamer uses a fixed entropy scale of $\eta = 3 \times 10^{-4}$ by normalizing returns to be approximately contained in the interval $[0, 1]$. We have the surrogate loss function:

$$\mathcal{L}(\phi) := -\sum_{t=1}^{T} \mathrm{sg}\Big((R_t^\lambda - v_\psi(s_t))/\max(1, S)\Big)\, \log \pi_\theta(a_t \mid s_t) \;+\; \eta\, H\big[\pi_\phi(a_t \mid s_t)\big]. \tag{3}$$

All three components are trained concurrently from replayed experience.

3.5 Teacher-Student Knowledge Distillation

Our contribution to the body of research is to introduce a new student model trained end-to-end on non-privileged observations o_t provided by CARLA, aided by the help of a pre-trained teacher model.

Our student model is exactly the DreamerV3 architecture used in CarDreamer, with the modification of removing BEV maps and intention sharing. We strip the BEV map of everything except waypoints for navigation, such that our BEV map consists only of a rectangular bounding box for the ego driver and a series of dots given by the CARLA waypoint manager.

We denote the pre-trained model weights by ϕ and the student weights by ψ. The pre-trained agent uses a DreamerV3 world model to produce latent states:

$$\textbf{Encoder:} \quad z_t \sim q_\phi(z_t \mid h_t, x_t),$$

The student model then encodes non-privileged observations using the same encoder structure.

$$\textbf{Student Encoder:} \quad \overline{z}_t \sim q_\psi(\overline{z}_t \mid h_t, o_t),$$

We retrain an end-to-end DreamerV3 model on environmental info with additional latent-matching loss given by KL divergence between latent states:

$$\mathcal{L}_{\text{distill}}(\psi) := \max\Big(1, \mathrm{KL}\Big[q_\psi(\overline{z}_t \mid h_t, o_t) \,\Big\|\, \mathrm{sg}\left(q_\phi(z_t \mid h_t, x_t)\right)\Big]\Big) \tag{4}$$

We initialize the student's weights by the teacher $\psi \leftarrow \phi$ except for input dimension on encoder CNNs and MLPs (since input dimensions differ). For unique information provided to the student model like non-privileged BEVs and increased camera angles we randomly initialize weights, and for privileged input information p_t in the teacher model we disregard the model weights.

Given an added hyperparameter $\beta_{\text{distill}} := 1$, the total student world-model loss becomes

$$\mathcal{L}_{\text{world}}(\psi) := \mathbb{E}_{q_\psi}\left[\sum_{t=1}^{T}\left(\beta_{\text{pred}}\mathcal{L}_{\text{pred}}(\psi) + \beta_{\text{dyn}}\mathcal{L}_{\text{dyn}}(\psi) + \beta_{\text{rep}}\mathcal{L}_{\text{rep}}(\psi) + \beta_{\text{distill}}\mathcal{L}_{\text{distill}}(\psi)\right)\right]. \quad (5)$$

3.6 Training Details

We report the environmental input visualization in Fig 2, the training logs in Fig 4, and some basic driving statistics (collision, speed, and distance) in Fig 5. We can see that throughout the episodes, the rewards increase, followed by a lower collision rate and a higher speed and distance of the vehicle. Additionally, we observe that the student model first reaches the destination at step 2896.

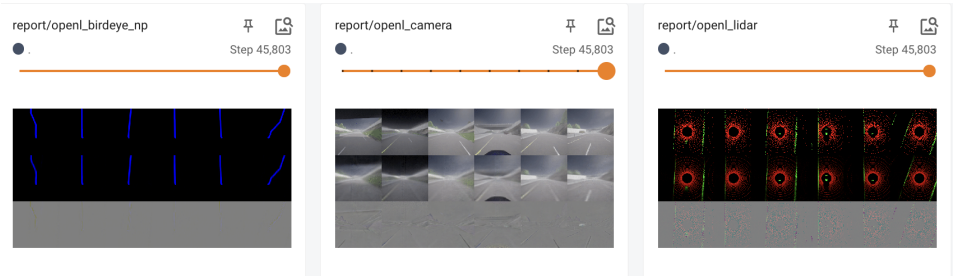

Figure 2: The environmental inputs of the model. We provide the birdeye images (left), the camera inputs (middle), and lidar (right).

4 Results and Analysis

We evaluate the Student-Teacher model against a baseline model trained without privileged information on three tasks: Left Turn Simple, Four Lane, and Navigation. Below, we summarize performance differences, focusing on task completion, safety, and control efficiency.

4.1 Left Turn Simple

The Student-Teacher model achieves **perfect task completion** (100.00% success) with **no collisions or out-of-lane events**, while the Baseline completes only 32.29% of runs and leaves the lane 67.70% of the time. The Student-Teacher model also covers more distance and maintains a higher average speed. Although its maximum speed is slightly lower, this may reflect improved control rather than conservatism.

Metric	Baseline	Student-Teacher
Success rate	32.29%	100.00%
Collision rate	0.00%	0.00%
Out-of-lane rate	67.70%	0.00%
Distance traveled	48.79	60.88
Avg. mean speed (normalized)	2.87	3.45
Avg. max speed (normalized)	5.25	4.89

Table 1: Evaluation results for the Left Turn Simple task.

4.2 Four Lane

On this more complex task, the Student-Teacher model again demonstrates superior performance with a **success rate of 92.31%** compared to 25.23% for the Baseline. While both models avoid lane violations, the Student-Teacher model travels farther and sustains a higher average speed. A slightly higher collision rate and slightly lower peak speed suggest more aggressive but smoother driving.

Metric	Baseline	Student-Teacher
huSuccess rate	25.23%	92.31%
Collision rate	6.54%	7.69%
Out-of-lane rate	0.00%	0.00%
Distance traveled	133.43	150.36
Avg. mean speed (normalized)	2.98	3.57
Avg. max speed (normalized)	5.50	4.93

Table 2: Evaluation results for the Four Lane task.

4.3 Navigation

Navigation is one of the hardest tasks provided by CarDreamer, especially due to the length of the destination. Therefore, even the original CarDreamer model has a near 0% rate of reaching the destination. However, judging by collision rate and distance traveled, our student model achieves a performance comparable to that of the teacher model.

Metric	Teacher	Student
Collision rate	6.62%	7.33%
Distance traveled	204.10	198.72

Table 3: Evaluation results for the Navigation task.

4.4 Analysis

As shown in the results above, the Student-Teacher framework significantly enhances performance. This highlights the challenge of learning directly from non-privileged information, such as raw visual inputs, which often lack precise cues about lane markings, object locations, or dynamics. A teacher model trained with privileged information helps the student learn what to attend to in the non-privileged input, effectively transferring structural knowledge without requiring privileged access at inference time.

This not only improves generalization but also aligns the student model with real-world deployment constraints, where privileged data (e.g., ground-truth maps or object labels) is unavailable. Separately, the use of a learned latent world model offers a practical advantage by bypassing the need to run computationally expensive simulations like CARLA during training. This reduces training cost and enables faster experimentation without sacrificing downstream performance.

5 Future Work

While our current results demonstrate that student models trained with latent-space alignment from privileged teacher models can achieve high success rates across simulated driving tasks, several important extensions remain. First, future work can evaluate this method on more challenging CARLA tasks such as Navigation with dynamic traffic, adverse weather, and longer time horizons. These will test generalization beyond structured intersections and expose limitations in robustness under uncertainty. Second, it would be interesting to incorporate stochastic sensory noise and partial observability into the student's input space to better simulate real-world deployment scenarios. Third, future work can extend the student-teacher framework to multi-agent settings, where cooperative or adversarial vehicles influence driving behavior, enabling the exploration of social driving competencies. Lastly, while our KL-based latent distillation proved effective, a future direction could be to investigate more expressive alignment techniques, such as contrastive representation learning or adversarial domain adaptation, to further narrow the sim-to-real gap.

6 Discussion

While our approach demonstrates strong sim-to-real transfer via latent-space alignment, it has several limitations. First, our experiments are restricted to CARLA scenarios that, while diverse, cannot fully capture the complexity and unpredictability of real-world driving. Second, the success of KL-based

distillation depends on the quality of the teacher model and may not generalize if the privileged model exhibits brittleness or bias. Additionally, our method assumes temporal synchronization between teacher and student representations, which may not hold under severe domain shifts. Nonetheless, the broader impact of this work lies in its potential to reduce reliance on costly or unrealistic supervision by leveraging structured simulation knowledge, ultimately accelerating the deployment of safer, more accessible autonomous driving systems trained with limited real-world data. One significant obstacle we met was a jax error when we adapted the skeleton code to our student-teacher framework, but we realized it was because of checkpoint incompatibility, so we ended up manually rewriting the keys of our trained checkpoint.

7 Conclusion

This work presents a practical and effective approach for bridging the sim-to-real gap in autonomous driving through latent-space alignment between world models trained with and without privileged information. Our student-teacher framework enables the transfer of high-level driving competencies from simulation-trained experts to models constrained to real-world sensory inputs, improving task success without requiring privileged data at inference. By leveraging latent distillation within a recurrent world model, we show that structure learned in simulation can generalize to more realistic conditions. These results highlight the promise of representation alignment for scalable, real-world deployment of autonomous agents, and provide a foundation for future advances in robust, sensor-based decision making in dynamic environments.

8 Team Contributions

- **Jack Liu**: methods formulation, software, RL implementation
- **Ryan Catullo**: methods formulation, software, RL implementation
- **Mac Ya**: methods formulation, software, RL implementation
- **Sunny Yu**:methods formulation, software, RL implementation

References

Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. 2019. Generating Adversarial Driving Scenarios in High-Fidelity Simulators. In *2019 International Conference on Robotics and Automation (ICRA)*. 8271–8277. https://doi.org/10.1109/ICRA.2019.8793740

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X Liu. 2021. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications* 12, 1 (2021), 748.

Dechen Gao, Shuangyu Cai, Hanchu Zhou, Hang Wang, Iman Soltani, and Junshan Zhang. 2024. Cardreamer: Open-source learning platform for world model based autonomous driving. *IEEE Internet of Things Journal* (2024).

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104* (2023).

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2024. Mastering Diverse Domains through World Models. arXiv:2301.04104 [cs.AI] https://arxiv.org/abs/2301.04104

Amir Khosravian, Masoud Masih-Tehrani, and Abdollah Amirkhani. 2022. The Semantic Segmentation of Autonomous Vehicles Images with the Teacher-Student Technique. *Electronic and Cyber Defense* 9, 4 (2022), 1–19.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems* 23, 6 (2021), 4909–4926.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. 2024. Think2Drive: Efficient Reinforcement Learning by Thinking with Latent World Model for Autonomous Driving (in CARLA-V2). In *European Conference on Computer Vision*. Springer, 142–158.

Wenjing Li, Jing Wang, Tingting Ren, Fang Li, Jun Zhang, and Zhongcheng Wu. 2022. Learning accurate, speedy, lightweight CNNs via instance-specific multi-teacher knowledge distillation for distracted driver posture identification. *IEEE transactions on intelligent transportation systems* 23, 10 (2022), 17922–17935.

Błażej Osiński, Adam Jakubowski, Paweł Zięcina, Piotr Miłoś, Christopher Galias, Silviu Homoceanu, and Henryk Michalewski. 2020. Simulation-based reinforcement learning for real-world autonomous driving. In *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6411–6418.

Xinlei Pan, Yurong You, Ziyan Wang, and Cewu Lu. 2017. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952* (2017).

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532* (2017).

Ruoyu Song, Muslum Ozgur Ozmen, Hyungsub Kim, Raymond Muller, Z Berkay Celik, and Antonio Bianchi. 2023. Discovering adversarial driving maneuvers against autonomous vehicles. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2957–2974.

A CarDreamer Details

Metric	Value
Episode Length	474
Episode Score	1127.92
Destination Reached	1
Collision Events	0
Out-of-Lane Events	0
Time Exceeded Events	0
Travel Distance (m)	157.41
Mean Travel Distance (normalized)	0.34
Max Travel Distance (normalized)	0.47
Mean Speed (normalized)	3.40
Max Speed (normalized)	4.72
Mean Waypoint Distance	0.90
Max Waypoint Distance	1.47
Mean Time-To-Collision (s)	2.49
Max Time-To-Collision (s)	15.52
Step Duration (s)	22.11
Frames Per Second (FPS)	21.48

Table 4: The evaluation result of the CarDreamer checkpoint evaluated on the Four Lane task provided in Gao et al. (2024) on 50,000 steps, which we use as a baseline result to compare with our trained model.

Quantitative Results: Table 4 shows the evaluation of the baseline CarDreamer model after 50,000 environment steps on the Four-Lane benchmark:

- The model achieved a perfect task completion rate (Destination Reached = 1) with **zero collisions, off-lane, or time exceedance events**, indicating high safety and stability.
- The normalized mean speed was 3.40 with a max of 4.72, showing reasonable control of speed.
- The average Time-To-Collision (TTC) was 2.49 seconds, with a max TTC of 15.52s, suggesting good anticipation and safe buffer times.
- Travel distance and waypoint adherence metrics show that the agent follows the designated path reliably.

B Evaluation Details

B.1 Baseline

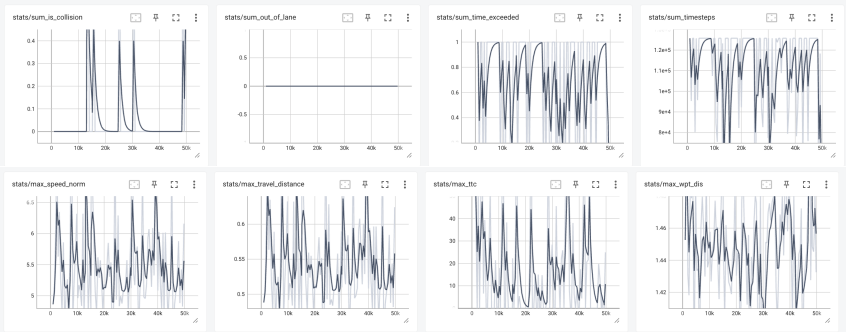

Figure 3: Evaluation results for our student-teacher model. (for Four Lane)

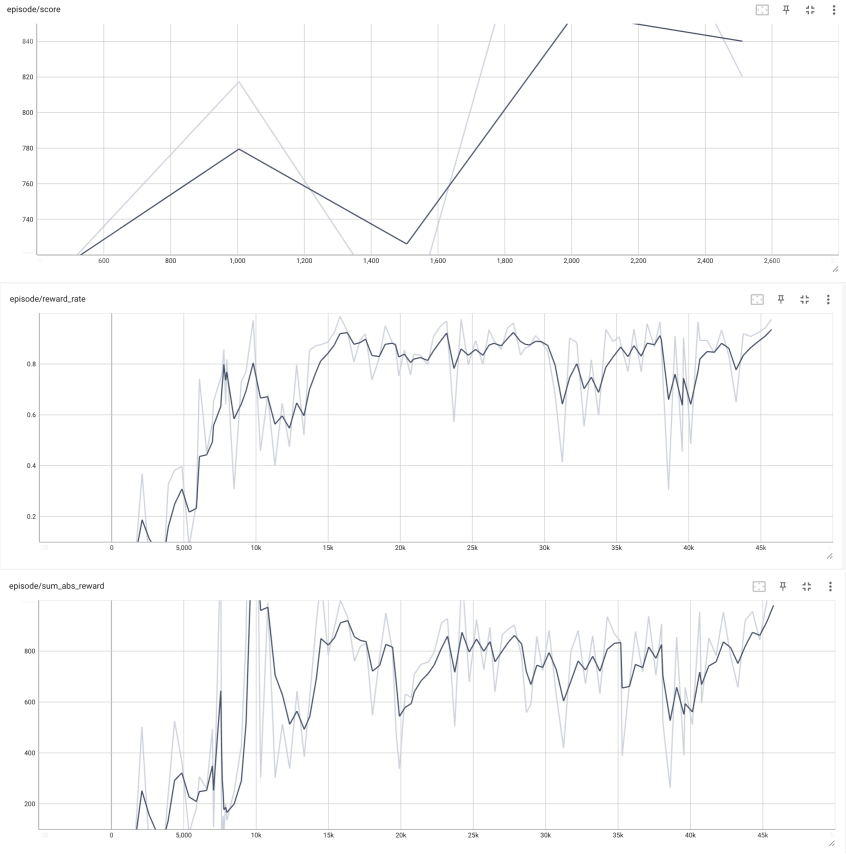

Figure 4: The training logs of score (up), reward rate (middle), and absolute reward value (down). (for Four Lane)

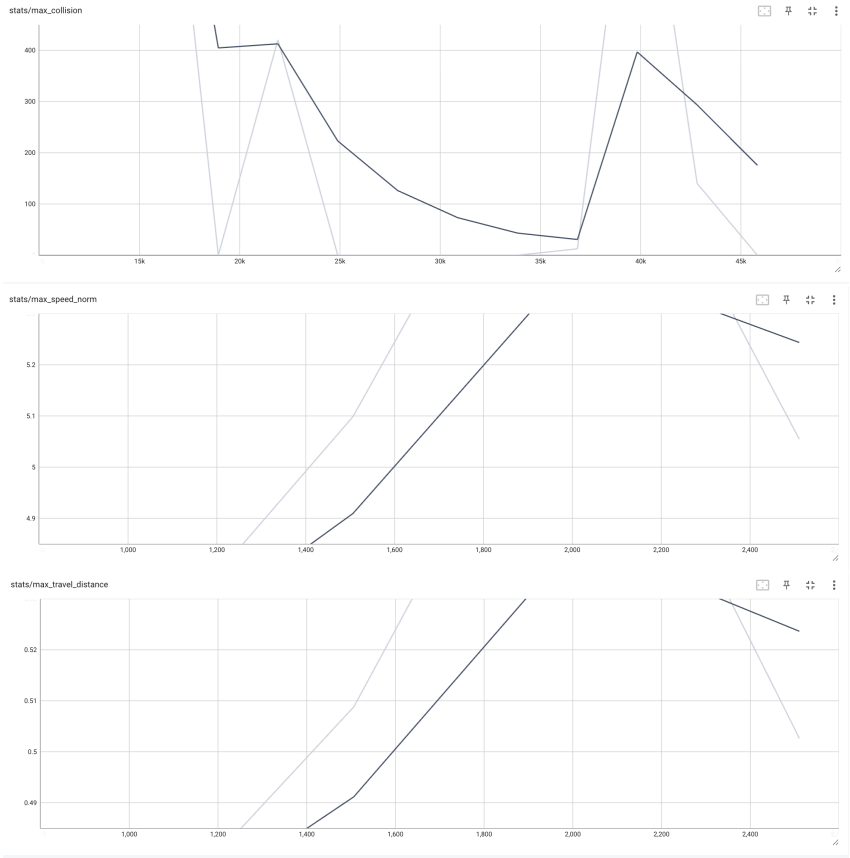

Figure 5: The collision rate (up), speed (middle), and distance traveled (down) during training. We can see that collision rate decreases while the speed and distance traveled both increase. We observe that earlier in the training process the student model first does not drive at all, then drives very slowly until being pushed out of lane by surrounding cars. Eventually, the model learns to drive in the middle of the lane, at a speed that's similar to other vehicles in the simulation. (for Four Lane)

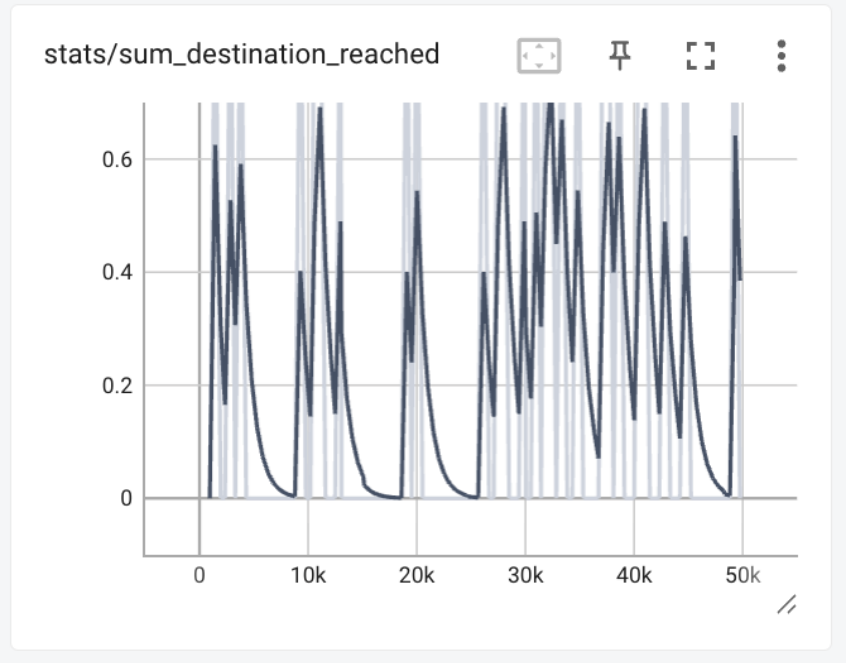

Figure 6: The destination-reached rate of our student-teacher model. Unlike the baseline model that never reaches the destination, our student-teacher model reached a higher success rate for Four Lane.