Closed-Loop Evaluation of Bird's-Eye-View Maps from Cross-View Transformers as Inputs to Behavior-Cloning Policies

Felipe Carlos dos Santos[1], **Eric Aislan Antonelo**[1], **Gustavo Claudio Karl Couto**[1]

[1] Federal University of Santa Catarina (UFSC), Florianópolis, Brazil
Department of Automation and Systems Engineering

felipe.carlos.santos@posgrad.ufsc.br,

eric.antonelo@ufsc.br, gustavo.karl.couto@posgrad.ufsc.br

Abstract. *In autonomous driving, Bird's-Eye View (BEV) representations provide a structured, top-down abstraction of the vehicle's surroundings and have become a key input modality for Behavioral Cloning (BC) policies. While ground-truth BEV maps are readily available in simulation, real-world deployment requires replacing them with camera-predicted counterparts—a substitution that introduces perceptual errors whose downstream impact on closed-loop driving performance is not well understood. In this work, we investigate the use of Cross-View Transformer (CVT)-predicted BEV maps as direct policy inputs for a BC agent in the CARLA simulator. We propose a six-channel BEV representation covering road surface, planned route, lane boundaries, vehicles, pedestrians, and traffic lights, and introduce a Kernel Density Estimation (KDE) weighting scheme that rebalances the segmentation loss towards underrepresented driving maneuvers such as curves and intersections. Closed-loop evaluation across two CARLA towns shows that the KDE-weighted model is the only predicted-BEV agent to complete a full episode without infractions, despite not achieving the highest aggregate IoU. This discrepancy reveals that global segmentation metrics are poor proxies for driving performance: what determines navigation success is prediction quality at geometrically critical locations, and the route channel emerges as the primary bottleneck for reliable agent navigation under predicted BEV inputs.*

1. Introduction

Autonomous driving has traditionally relied on modular pipelines that sequentially process perception, planning, and control. However, learning-based approaches have gained significant traction for their ability to capture complex driving behaviors directly from data, often leveraging Behavior Cloning (BC) trained on offline expert demonstrations [Bojarski et al. 2016, Bansal et al. 2018, Codevilla et al. 2017]. A critical enabler for these methods is the Bird's-Eye View (BEV) representation, which removes perspective distortions from ego-centric cameras and exposes scene geometry in a planner-friendly grid [Philion and Fidler 2020]. Recent transformer-based architectures learn multi-camera view transformations implicitly: Cross-View Transformers (CVT) [Park et al. 2023] introduced camera-aware cross-view attention for real-time semantic BEV generation from onboard cameras alone, without requiring LiDAR or dense depth supervision, making BEV prediction feasible in camera-only, single-vehicle settings.

Decoupling perception from policy learning through structured BEV representations simplifies the control task and improves sample efficiency [Ota et al. 2021, Couto and Antonelo 2023]. Behavior cloning (BC) from limited expert demonstrations can be effectively conditioned on BEV representations and sparse route inputs for simulated city navigation [Antonelo et al. 2024], validating BEV as a compact, semantically rich interface between perception and control. Preliminary studies have explored CVT-based BEV generation under single-vehicle configurations and limited training towns [Santos et al. 2025], yet the downstream impact of camera-only predicted BEVs on closed-loop driving performance remains not adequately investigated.

This work proposes BEV generation models trained to predict six one-hot encoded semantic channels: roads, planned route, lane boundaries, vehicles, pedestrians, and traffic lights. To assess the practical utility of these representations, the generated BEVs are integrated into a BC driving policy and their downstream impact on autonomous navigation is evaluated. As an initial benchmark, the proposed models are compared against the three-channel baselines [Santos et al. 2025] in a simulated driving environment without dynamic obstacles, enabling an isolated analysis of the agent's ability to follow routes and execute driving maneuvers using predicted BEV representations. The main contributions of this work are:

- A six-channel CVT-based BEV generation model trained on a large, perturbation-augmented dataset, with six channels in contrast to the three-channel model of [Santos et al. 2025], encoding roads, route, lane boundaries, vehicles, pedestrians, and traffic lights.
- A KDE-based importance-weighting scheme that rebalances the BEV segmentation loss towards underrepresented driving maneuvers, such as curves and intersections, improving route and road channel quality at geometrically critical locations.
- A closed-loop evaluation of multiple BEV generation models as policy inputs in the CARLA simulator.

2. Methods

2.1. Cross-view transformer (CVT)

CVT [Park et al. 2023] processes a set of n monocular views $\{(I_k, K_k, R_k, t_k)\}_{k=1}^{n}$, where $I_k \in R^{H\times W\times 3}$ is the image, $K_k \in R^{3\times 3}$ the intrinsic matrix, and (R_k, t_k) the extrinsic rotation and translation relative to the ego-vehicle. Images are encoded by an EfficientNet-B4 [Tan and Le 2019] backbone, and camera-aware positional encodings derived from per-view projection geometry are fused with image features via cross-view attention, aggregating multi-view information into a unified BEV latent space. The map-view representation is iteratively refined at multiple resolutions before being upsampled to the final prediction. Full details on the attention formulation and positional encoding can be found in [Park et al. 2023].

The CVT output is a six-channel 192×192 top-down map, where each channel is a binary segmentation mask for one semantic class: road surface, planned route, lane boundaries, vehicles, pedestrians, and traffic lights.

2.2. Behavior Cloning (BC)

Behavior Cloning (BC) trains a policy π_θ on a dataset $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^{N}$ of expert demonstrations collected in CARLA, where s_i pairs a ground-truth BEV image with the vehicle's

current speed and previous control commands, and $a_i = (a_i^{\text{acc}}, a_i^{\text{steer}}) \in [-1, 1]^2$ is the expert's acceleration and steering [Pomerleau 1988]. Given s_i, the network outputs four scalars $(\alpha^{\text{acc}}, \beta^{\text{acc}}, \alpha^{\text{steer}}, \beta^{\text{steer}})$ that parameterize independent Beta distributions; at inference, control actions are sampled from these distributions [Petrazzini and Antonelo 2021, Antonelo et al. 2024]. The bounded support of $\mathcal{B}(\alpha, \beta)$ on $[0, 1]$ naturally accommodates actuator constraints without post-hoc clipping. Actions on $[-1, 1]$ are rescaled via $a' = (a + 1)/2$, and the training objective is the mean negative log-likelihood:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \left[\log f\left(\tfrac{a_i^{\text{acc}}+1}{2};\ \alpha_i^{\text{acc}}, \beta_i^{\text{acc}}\right) + \log f\left(\tfrac{a_i^{\text{steer}}+1}{2};\ \alpha_i^{\text{steer}}, \beta_i^{\text{steer}}\right) \right], \tag{1}$$

where $f(x; \alpha, \beta) = x^{\alpha-1}(1 - x)^{\beta-1}/B(\alpha, \beta)$ is the Beta density and $B(\alpha, \beta)$ is the Beta function.

2.3. Kernel Density Estimation for Segmentation Training

A Kernel Density Estimator (KDE) is a non-parametric method to estimate the probability density function (PDF) of a random variable. For each data point in the sample, a kernel function is centered at that point. The KDE is computed by summing all these kernel functions and normalizing by the number of data points. Mathematically, the KDE for a point x is given by:

$$\hat{g}(x) = \frac{1}{Nh} \sum_{j=1}^{N} K\left(\frac{x - x_j}{h}\right), \tag{2}$$

where $\hat{g}(x)$ is the estimated density at point x, N is the number of data points, h is the bandwidth, x_j are the data points, and K is the kernel function.

The Gaussian kernel is commonly used in KDE as it assigns weights to data points based on their distance from the estimation point, ensuring that closer points exert a greater influence on the density estimate. It is defined as:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}},$$

where u is the standardized distance between the estimation point and the data point, scaled by the bandwidth h.

In this work, KDE is applied to reweight training samples during the optimization of one of our studied BEV segmentation models. The KDE is fitted over the trajectory actions (steering angle and throttle) recorded in the training set, providing a density estimate of the action distribution. By using the inverse of this density as a per-sample weight, the model is encouraged to pay greater attention to underrepresented driving scenarios, such as sharp turns and intersections, which are the most challenging and least frequent contexts in the training data.

Specifically, the per-sample weight w_i is defined as:

$$w_i = \frac{1/\hat{g}(a_i)}{\frac{1}{N}\sum_{j=1}^{N} 1/\hat{g}(a_j)}, \tag{3}$$

where $\hat{g}(a_i)$ is the kernel density estimate evaluated at the action a_i of sample i, and the denominator normalizes the weights to have unit mean across the training set.

The per-sample segmentation loss ℓ_i is the binary cross-entropy (BCE) averaged over all C output channels and the spatial dimensions $H \times W$ of the BEV map:

$$\ell_i = \frac{1}{C \cdot H \cdot W} \sum_{c=1}^{C} \sum_{h,w} \mathrm{BCE}(\hat{y}_{i,c,h,w},\ y_{i,c,h,w}) \,, \tag{4}$$

where $\hat{y}_{i,c,h,w}$ is the model prediction and $y_{i,c,h,w}$ is the ground-truth binary mask for channel c at spatial position (h, w).

The resulting KDE-weighted training objective is:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \ell_i. \tag{5}$$

This formulation ensures that samples from rare driving situations contribute more strongly to the gradient, thereby biasing the model towards better performance on the scenarios where accurate BEV prediction matters most.

3. BC Policy Agent

The BC policy agent takes a BEV image and scalar vehicle state as input, and outputs continuous control actions via a Beta distribution head. The following subsections detail the input representation, action space, and network architecture.

3.1. Input representation

The policy agent receives as input a 192×192 three-channel Bird's-Eye View (BEV) image produced by the perception network. In the present evaluation, only the first three semantic channels are employed, corresponding to the road surface, planned route, and lane-boundary representations (channels 0–2). This restriction is motivated by three factors: it enables direct comparison with the three-channel baselines from [Santos et al. 2025]; the KDE weighting strategy described in Section 2.3 improves prediction quality primarily for these static geometry channels, since they are most affected by the underrepresented driving maneuvers the KDE targets; and channels 3–5 (vehicles, pedestrians, and traffic lights) are severely class-imbalanced relative to channels 0–2: whereas road, route, and lane-boundary pixels cover large and consistent fractions of each frame, dynamic elements occupy sparse, intermittent regions, posing a representational challenge that translates directly into near-zero IoU across all trained models, as further discussed in Section 4 (Table 2); their improvement is left for future work.

In addition to the visual input, the agent receives the current longitudinal speed of the ego-vehicle along with the most recent policy actuator outputs, namely the previous acceleration and steering commands. These scalar inputs are concatenated with the feature representation at a later stage of the network, specifically at the first fully connected layer.

3.2. Output Representation

The action space is $\mathbf{a} \in [-1, 1]^2$, comprising steering and a unified acceleration signal that encodes braking as negative acceleration, preventing simultaneous throttle and brake commands [Petrazzini and Antonelo 2021]. The policy output is modeled by a Beta distribution as described in Section 2.2.

3.3. Network architecture

The BC agent employs the same network architecture used in [Antonelo et al. 2024]. The BEV image is processed through a stack of convolutional layers, producing a spatial embedding that is concatenated with a state embedding obtained by passing the scalar vehicle state variables through a small MLP. The fused representation is then projected to a 256-dimensional feature vector, which is mapped by a subsequent MLP policy head to the parameters of a Beta distribution, from which the continuous control actions, acceleration $a^{\text{acc}} \in [0, 1]$ and steering $a^{\text{steer}} \in [-1, 1]$, are sampled.

4. Experiments

In this section, we evaluate three CVT models—*CVT 6ch. No Noise*, *CVT 6ch.*, and *CVT 6ch. KDE*—first via per-channel IoU for BEV quality, then via closed-loop driving in CARLA to assess the downstream impact of predicted BEV on autonomous navigation.

4.1. Dataset Generation

The simulation environment and route definitions are sourced from the CARLA Leaderboard evaluation platform [lea 2020]. Specifically, the *Town01* environment is employed for dataset collection, comprising ten predefined routes used to build the expert demonstration dataset, while *Town02* is reserved exclusively for testing purposes. Each town provides ten distinct routes.

Compared to the dataset from [Santos et al. 2025], a larger and more diverse dataset is proposed here, as the inclusion of dynamic agents, vehicles, pedestrians, and traffic lights, substantially increases the complexity of the driving task. The dataset consists of five independent collections of the ten routes available in *Town01*. Across all five collections, route 00 was designated as the validation split during training, while routes 01 through 09 constituted the training set.

To increase data variability and simulate different levels of driving proficiency, steering and throttle perturbations were injected at varying magnitudes, ranging from large disturbances capable of displacing the vehicle into the opposing lane or off the road entirely, to zero-noise demonstrations. Additionally, each collection was generated using a distinct random seed, resulting in different combinations of vehicle and pedestrian appearances, shapes, and spawn configurations across collections.

In total, the dataset comprises 1,186,499 images and 201,980 sample points (≈93.8 GB):

- **Training:** 930,040 images, 155,007 points (78.2 GB).
- **Validation:** 129,579 images, 21,597 points (10.5 GB).
- **Test:** 126,880 images, 25,376 points (5.1 GB).

Compared to the dataset from [Santos et al. 2025], the training and validation sets grew approximately 10× (from 106,435 to 1,059,619 images; from 4.1 GB to 88.7 GB), while the test set remains unchanged to preserve comparability with previously reported results.

4.2. Training

Three models were trained using the Binary Cross-Entropy (BCE) loss with six input channels: four RGB images captured from front, right, left, and rear-facing cameras, and one single-channel image representing a sparse route map, following the architecture described in Section 2.1. All networks presented in this paper were trained for 50 epochs, following the same setting used in [Santos et al. 2025], to ensure a uniform comparison across models.

CVT 6ch. No Noise uses the smaller, unperturbed dataset from [Santos et al. 2025]; the 3-channel vs. 6-channel distinction is relevant when comparing against that baseline. *CVT 6ch.* uses the same architecture on the larger dataset from Section 4.1. *CVT 6ch. KDE* adds the importance-weighting scheme from Section 2.3, assigning higher BCE loss weights to underrepresented curve and intersection samples.

The BC policy is trained exclusively with ground-truth BEV maps obtained directly from CARLA. This choice avoids retraining a separate policy for each BEV generation model, which would introduce an additional source of variability and make it harder to isolate the effect of BEV prediction quality on driving performance. At evaluation time, the ground-truth BEV is replaced by the output of each perception model, so that the only varying factor across runs is the BEV generator.

4.3. Evaluation Protocol

The evaluation is divided into two parts. First, BEV prediction quality is assessed using Intersection over Union (IoU) computed independently for each output channel, on the validation split of Town01 and on the held-out test set of Town02. Second, the driving performance of the BC agent is measured in closed-loop episodes within the CARLA simulator, using the generated BEV maps as policy input in place of the ground-truth maps used during training.

4.4. BEV Evaluation

Table 1 shows that the six-channel BEV task splits into two sub-problems: channels 0–2 (roads, route, lane boundaries) encode static geometry present in every frame, while channels 3–5 (vehicles, pedestrians, traffic lights) require dynamic obstacle recognition, a harder task given element sparsity in the data.

For channels 0–2, KDE weighting provides a clear benefit: *CVT 6ch. KDE* achieves the highest mean IoU (0.6413) among the larger-dataset models, outperforming *CVT 6ch.* (0.5897). Since curve and intersection samples receive higher loss weight, and these geometries directly determine road, route, and lane-boundary quality, the improvement is consistent with the KDE motivation. *CVT 6ch. No Noise*, trained on the smaller unperturbed dataset from [Santos et al. 2025], achieves only 0.2455, confirming that dataset scale and perturbation diversity are key drivers of segmentation quality independently of KDE weighting.

For the obstacle-related channels (3–5), however, the benefit of KDE weighting is not observed. The mean IoU for channels 3–5 is comparable between *CVT 6ch.* (0.0997) and *CVT 6ch. KDE* (0.0967), with the KDE variant even showing a marginal decrease relative to the vanilla model. This suggests that reweighting samples based on the density

Table 1. Mean per-channel IoU — City 1 (Town01, validation set).

Channel	CVT 6ch.	CVT 6ch. No Noise	CVT 6ch. KDE
Roads	0.8135	0.5086	**0.8614**
Trajectory	0.5938	0.1918	**0.6455**
Lane	0.3618	0.0362	**0.4170**
Vehicles	**0.2460**	0.0696	0.2401
Pedestrians	0.0000	0.0000	0.0000
Traffic Light	**0.0532**	0.0031	0.0499
Mean (Ch. 0–2)	0.5897	0.2455	**0.6413**
Mean (Ch. 3–5)	**0.0997**	0.0242	0.0967

Bold values indicate the best score per row.

of steering and throttle actions, which is informative for the static road geometry task, does not transfer to the detection of vehicles, pedestrians, and traffic lights, whose occurrence is governed by factors largely independent of the ego-vehicle's own control actions. Addressing this limitation is left for future work.

The results on the test set, summarized in Table 2, corroborate these findings in an unseen town. The *CVT 6ch. KDE* model again achieves the best mean IoU for channels 0–2 (0.5667) among the models trained on the proposed dataset, surpassing both *CVT 6ch.* (0.5519) and *CVT 6ch. No Noise* (0.4459), while remaining below the *CVT 3ch.* baseline (0.6703). For channels 3–5, the pattern observed in Town01 persists, with *CVT 6ch.* and *CVT 6ch. KDE* performing similarly (0.0887 and 0.0852, respectively), reinforcing that the advantage of KDE weighting is confined to the road-geometry sub-task and does not generalize to obstacle recognition.

It is worth noting, however, that route-averaged IoU is inherently biased towards straight segments, which dominate route length, while curves and intersections—driving-critical locations—account for a small fraction of the total. This explains why aggregate metrics favor *CVT 3ch.* (Table 4), yet this advantage does not translate into better closed-loop driving performance, as discussed in the next section.

4.5. BC Agent Evaluation

All BEV generation models are evaluated by substituting the ground-truth BEV with their respective predictions at inference time, while keeping the policy weights fixed. To isolate the effect of BEV prediction quality on driving performance, the evaluation environment contains no dynamic agents: other vehicles and pedestrians are absent, and traffic lights are disabled. Under these conditions, the policy driven by the ground-truth BEV successfully completes all five episodes in both towns without any infraction (Table 3), establishing a strong upper-bound reference for the evaluation.

Each model was evaluated in five fixed-spawn closed-loop episodes per town, each defining a route of approximately 1,241 m; episodes end on route completion or upon an infraction. Figure 1 overlays the resulting trajectories on the CARLA top-down maps: colored paths trace each model's route, arrows mark infraction positions, and numbered markers link episode start and end points.

Table 2. Mean per-channel IoU — City 2 (Town02, test set).

Channel	CVT 6ch.	CVT 6ch. No Noise	CVT 6ch. KDE	CVT 3ch.†	UNet†
Roads	0.7889	0.6810	0.8160	**0.9144**	0.6978
Trajectory	0.5569	0.4082	0.5755	**0.7808**	0.5915
Lane	0.3098	0.2485	0.3086	0.3158	**0.3959**
Vehicles	**0.2419**	0.1221	0.2252	–	–
Pedestrians	0.0001	0.0000	**0.0003**	–	–
Traffic Light	0.0240	0.0051	**0.0302**	–	–
Mean (Ch. 0–2)	0.5519	0.4459	0.5667	**0.6703**	0.5617
Mean (Ch. 3–5)	**0.0887**	0.0424	0.0852	–	–

Bold values indicate the best score per row.
† Models trained with only 3 output channels in [Santos et al. 2025].

Table 3 summarizes the route completion distances and infraction counts for each model across both towns. Among all models that rely on predicted BEV representations, *CVT 6ch. KDE* achieves the best overall driving performance, accumulating 261 m of route completion in Town01 and 106 m in Town02. More notably, it is the only perception model capable of completing an entire episode without triggering any infraction, recording four infractions out of five episodes in Town01 rather than the five out of five observed for every other predicted-BEV model.

As shown in Figure 1, the primary failure modes of the evaluated agents occur at curves and road intersections. These are precisely the scenarios where KDE-based weighting proves beneficial. By assigning greater importance to training samples associated with turning maneuvers and intersection navigation, *CVT 6ch. KDE* learns more robust representations for these underrepresented situations, resulting in more reliable BEV predictions and improved route completion performance.

Table 3. Route completion (meters) and number of infractions out of 5 episodes for Town01 and Town02 (route length $\approx$ 1,241 m per town).

	Town01		Town02	
BEV Generator	Route (m)	Infrac.	Route (m)	Infrac.
Ground-truth BEV	*598*	*0/5*	*604*	*0/5*
UNet	115	5/5	22	5/5
CVT 3ch.	159	5/5	27	5/5
CVT 6ch.	151	5/5	52	5/5
CVT 6ch. KDE	**261**	**4/5**	**106**	**5/5**
CVT 6ch. No Noise	120	5/5	28	5/5

The episode successfully completed by *CVT 6ch. KDE* in Town01 corresponds to the second episode in Figure 1(a). This route requires the agent to navigate through at least one curve and two intersections before reaching the finish point. The fact that *CVT 6ch. KDE* completes this episode while all other models fail underscores a key finding: a

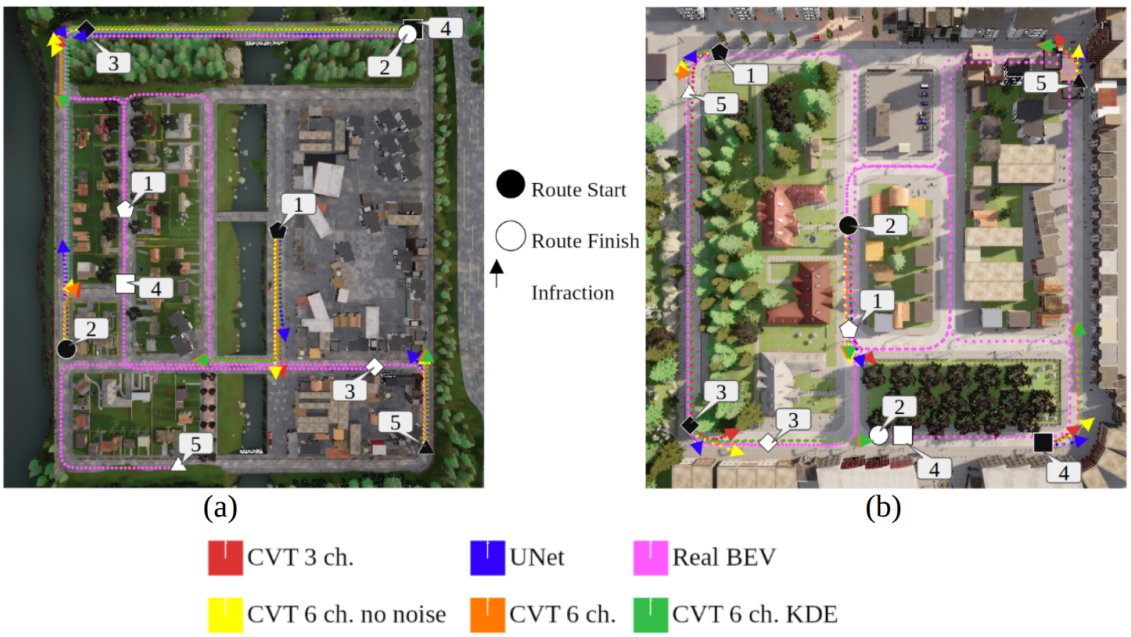


Figure 1. Trajectories generated by each evaluated model overlaid on the CARLA top-down maps for (a) Town01 and (b) Town02. Colors denote different BEV generation approaches, as indicated in the legend. Black filled markers represent episode starting locations, while white filled markers indicate successfully completed episodes. Colored arrows mark the vehicle position and heading at the time of an infraction that caused episode termination. The numerical labels identify the corresponding episode start and end points.

higher IoU mean on the evaluation routes does not necessarily translate into better closed-loop driving performance. As shown in Table 4, *CVT 3ch.* achieves the highest mean IoU in both towns (0.764 in Town01 and 0.679 in Town02), yet it fails all five episodes in each town with the shortest average route completion. Conversely, *CVT 6ch. KDE* ranks second in mean IoU (0.713 in Town01) but delivers the strongest driving results. This discrepancy suggests that the KDE weighting scheme, by up-weighting training samples associated with curves and intersections, produces a BEV representation that is more reliable precisely at the critical decision points where the policy is most likely to commit an infraction, even if this does not manifest as a global improvement in aggregate IoU metrics.

Table 4. Mean IoU per channel on the spawn evaluation for Town01 and Town02. Channels: Road (ch. 0), Route (ch. 1), Lane boundary (ch. 2).

	Town01				Town02			
BEV Generator	Road	Traj.	Lane	**Mean**	Road	Traj.	Lane	**Mean**
CVT 3ch.	0.933	**0.808**	**0.551**	**0.764**	0.871	**0.727**	**0.438**	**0.679**
CVT 6ch.	0.935	0.788	0.332	0.685	**0.903**	0.680	0.321	0.635
CVT 6ch. KDE	**0.943**	0.789	0.406	0.713	0.902	0.667	0.340	0.636

A qualitative analysis of the BEV predictions provides further insight into the

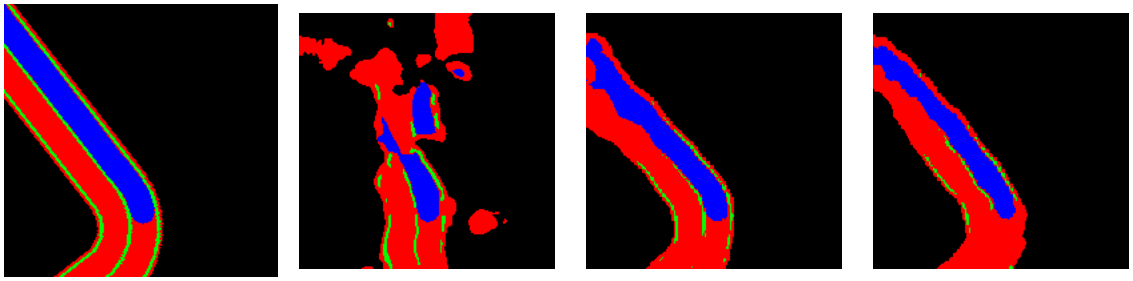


Figure 2. BEV predictions at the northwest curve of *Town01* near the start of episode three, where the road transitions from an east–west straight to a southward turn. predictions at the same timestep. Only the CVT 6ch. and CVT 6ch. KDE agents complete this curve.

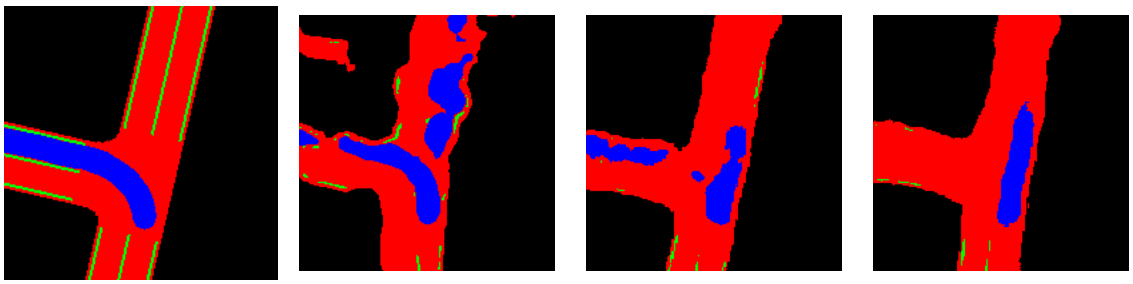


Figure 3. BEV predictions at the cross road in the middle of *Town02* near the finish point of episode one. predictions at the same timestep. All agents fail in this intersection.

driving failures observed in Table 3. Figure 2 illustrates a curve in Town01 where only *CVT 6ch.* and *CVT 6ch. KDE* produce a route channel that correctly traces the curvature ahead, allowing those agents to negotiate the turn, while *CVT 3ch.* fails to represent the curve and the agent consequently cannot initiate the maneuver. The fact that both six-channel models succeed while *CVT 3ch.* fails points to the larger and more diverse training dataset as the primary factor: steering and throttle perturbations expose both models to a wider range of trajectories through curves and off-nominal road positions. At this snapshot, the route channel (ch. 1) IoU is 0.586 for *CVT 6ch. KDE* and 0.669 for *CVT 6ch.*, a $\approx$12% difference that nonetheless leaves both predictions sufficient to guide the policy through the turn. The KDE weighting shows a stronger effect on the lane-boundary channel (ch. 2), where its IoU of 0.203 represents a $\approx$283% increase over the vanilla model's 0.053, illustrating that the KDE's benefit at this location is complementary rather than the decisive factor.

Figure 3 shows a left-turn intersection in Town02 where all agents fail. The better-performing models produce a route channel that tends to continue straight rather than tracing the intended left turn; *CVT 6ch. KDE* yields the closest approximation among them, yet its prediction remains insufficient to signal the required direction change to the policy.

Another qualitative example is shown in Fig. 4, focusing on a left-turn intersection



Figure 4. BEV predictions at the cross road in the east of *Town02* when CVT 6ch. KDE and CVT 6ch. fail to make a left turn. Predictions at the same timestep.

in the eastern section of Town02, where at the time of the snapshot only *CVT 6ch. KDE* and *CVT 6ch.* were still active. Both agents fail to execute the left turn at this location, and the route channel predictions reveal a clear reason: the snapshot-level IoU for the route channel (channel1) is 0.156 for *CVT 6ch. KDE* and only 0.092 for *CVT 6ch.*, both values substantially below what would be needed to reliably convey the intended turning direction to the policy. The degraded route representation at this intersection means the agent receives an ambiguous or incorrect signal regarding the planned path, causing it to continue straight or commit an infraction rather than turn. This case reinforces the observation that low route channel accuracy at geometrically critical points, particularly intersections requiring directional changes, is a primary driver of episode terminations, and that improving BEV prediction quality at these locations remains the key challenge for enabling the policy to complete full routes under predicted BEV inputs. Given the critical role of the route channel in conveying directional intent to the policy, future work will focus specifically on improving its representation fidelity, as gains in this channel are expected to translate most directly into improved agent drivability.

5. Conclusion

The results presented in this work demonstrate that the KDE-weighted six-channel CVT model represents a meaningful step forward in BEV-based behavioral cloning for autonomous driving, being the only predicted-BEV agent capable of completing a full episode without infractions in the evaluated environment. The KDE weighting strategy proved effective in directing the model's learning capacity towards the most challenging and underrepresented driving scenarios, particularly curves and intersections, producing route and road channel predictions that are qualitatively more reliable. Nevertheless, the overall driving performance remains far below that of the ground-truth BEV baseline, indicating that the quality of the route channel prediction is still the primary bottleneck limiting closed-loop performance. Improving the fidelity of the route representation at intersections and sharp turns is therefore the most pressing direction for future work, as the current accuracy levels are insufficient to consistently guide the policy through directional maneuvers.

Once the route channel quality reaches a level that enables more reliable navigation of the road geometry, subsequent efforts should shift towards the obstacle-related channels, namely vehicles, pedestrians, and traffic lights, which were deliberately ex-

cluded from the scope of this evaluation. The near-zero IoU values observed for these channels across all models confirm that their prediction remains an open challenge, one that will become increasingly relevant as the evaluation is extended to environments with dynamic agents. In this broader context, the KDE weighting technique has demonstrated its value as a general mechanism for rebalancing the training signal towards infrequent but safety-critical scenarios, and exploring its application to the obstacle channels, potentially conditioned on the density of dynamic agent occurrences in the dataset rather than on ego-vehicle actions, represents a promising direction for future investigation.

References

(2020). CARLA autonomous driving leaderboard.

Antonelo, E. A., Couto, G. C. K., Möller, C., and Fernandes, P. H. (2024). Investigating behavior cloning from few demonstrations for autonomous driving based on bird's-eye view in simulated cities. In *BRACIS*, pages 155–168. Springer.

Bansal, M., Krizhevsky, A., and Ogale, A. (2018). Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars.

Codevilla, F., Müller, M., Dosovitskiy, A., López, A. M., and Koltun, V. (2017). End-to-end driving via conditional imitation learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9.

Couto, G. C. K. and Antonelo, E. A. (2023). Hierarchical generative adversarial imitation learning with mid-level input generation for autonomous driving on urban environments. *arXiv preprint arXiv:2302.04823*.

Ota, K., Jha, D. K., and Kanezaki, A. (2021). Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*.

Park, S., Hong, S., Kim, S., and Lee, K. (2023). Cross-view transformers for real-time map-view semantic segmentation. In *IEEE/CVF CVPR*.

Petrazzini, I. G. and Antonelo, E. A. (2021). Proximal policy optimization with continuous bounded action space via the beta distribution. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Philion, J. and Fidler, S. (2020). Lift, splat, shoot: A multicamera bird's-eye view for 3d object detection. In *European Conference on Computer Vision (ECCV)*.

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.

Santos, F. C. d., Antonelo, E. A., and Couto, G. C. K. (2025). An initial study of bird's-eye view generation for autonomous vehicles using cross-view transformers. *arXiv preprint arXiv:2508.12520*.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR.