

SkyDrive: Learning to Drive in a New City from Aerial Traffic Monitoring

Weijiang Xiong, Lan Feng, Alexandre Alahi, Nikolas Geroliminis *

¹École Polytechnique Fédérale de Lausanne
firstname.lastname@epfl.ch

Abstract

Autonomous driving has made remarkable progress through imitation learning with massive human demonstration data. However, a trained planner often degrades severely when applied to a new environment zero-shot, because of domain shifts in traffic regulations, road layout and driving behaviors. Therefore, adapting a trajectory planner to a new city typically requires resource-demanding local data collection with a vehicle sensor suite. In this work, we show that driving behavior can be learned from a scalable and efficient alternative. We introduce *SkyDrive*, a framework that utilizes drone-based traffic monitoring to provide efficient supervision for autonomous driving agents in a new environment. While vehicle-based data collection logs the ego and its surroundings, an aerial platform naturally observes many road users simultaneously over an extended field of view. As a result, every vehicle can be a data source with grounded driving behavior, effectively scaling up the amount of supervision. Based on 137 hours of aerial traffic monitoring footage, we extract 650K driving samples and construct a benchmark for trajectory planners and motion predictors. Zero-shot experiments with multiple models reveal significant cross-city domain gaps, but many of them can be alleviated by limited supervision from the sky, e.g., 30 minutes of monitoring per location. Our findings show that aerial traffic monitoring is an efficient and scalable data source for adapting autonomous driving systems in new cities. Data and code will be made publicly available.

Introduction

Learning to imitate human driving behavior has been the cornerstone of autonomous driving. Trained on sufficient and high-quality expert demonstrations, such as Argoverse, nuPlan, and the Waymo Open Dataset, modern trajectory planners can learn to navigate a vehicle under complex scenarios (Wilson et al. 2023; Caesar et al. 2021; Ettinger et al. 2021). However, the strong in-domain capability often does not guarantee reliable transfer to an unseen environment, e.g., a new city, due to the recognized out-of-distribution problem (Zhou et al. 2023). Since geographical distribution shifts can arise from various aspects, including landscape, road layout and traffic regulations, performance degradations are often

expected in object detection (Chen et al. 2018; Yang et al. 2021), motion prediction (Yao et al. 2024) and trajectory planning (Yasarla et al. 2025). Therefore, zero-shot transfer is prone to pronounced errors (Naeinian et al. 2026).

To tackle this challenge, a direct solution is to retrain the model after collecting and annotating enough driving logs with instrumented vehicles in the target city. But conducting such an ambitious project for every new city demands lots of resources, especially for covering the long-tailed distribution of driving scenarios. Fortunately, in addition to the ego vehicle, surrounding agents can also provide valuable planning supervision (Chen and Krähenbühl 2022), which greatly improves data efficiency. However, unclear driving intentions, fragmented trajectories and perception noise have become the major obstacles due to sensor range limitations and occlusions (Zhang and Ohn-Bar 2021). Inspired by this, we opt for an alternative solution with minimal occlusion, clear visibility and an extended field of view, i.e., aerial traffic monitoring (Fonod et al. 2025). Such a platform can simultaneously detect and track many vehicles within the view, and all visible vehicles can be turned into driving demonstrations, making it an efficient and scalable data source.

We introduce *SkyDrive*, a framework that constructs driving scenarios from aerial traffic monitoring. Building upon geo-referenced vehicle tracks, *SkyDrive* carefully selects agents as virtual egos and extracts driving scenes centered on them. The scenes are rasterized into ego-centric multi-view semantic images for trajectory planning, and exported as vectorized tracks for motion prediction, allowing diverse supervision grounded on real driving behaviors. The benchmark reveals pronounced zero-shot degradation under domain shifts and shows that a small subset of monitoring sessions can substantially improve target-domain performance.

In summary, this work makes the following contributions:

- A scalable and efficient pipeline that converts aerial traffic monitoring into localized driving supervision.
- A high-quality dataset with ~ 650 K real scenarios from 137.2 hours of aerial observations over 20 complex modern intersections.
- Detailed experiments on trajectory planning and motion prediction to demonstrate the zero-shot transfer challenge and the efficiency of aerial supervision.

*Corresponding author.

Related Work

Autonomous Driving

Motion prediction is an important block in earlier paradigms of autonomous driving with modularized perception, prediction and planning. AutoBot jointly encodes the motion of all agents and decodes map-consistent futures (Girgis et al. 2022). MTR designs learnable queries to represent different driving intentions (Shi et al. 2022). Wayformer applies early fusion for heterogeneous information, including traffic light, map, ego history and surrounding agents (Nayakanti et al. 2023). UniTraj evaluates them on multiple datasets and reveals significant cross-domain gaps (Feng et al. 2024).

Since modularized systems may be suboptimal for the ultimate goal due to accumulated errors (Chen et al. 2024a), most modern solutions train end-to-end models to learn trajectory planning directly from sensor observations. For example, TransFuser combines camera and LiDAR features for waypoint prediction (Prakash, Chitta, and Geiger 2021). The Bird’s-Eye-View (BEV) is a central latent space for fusing multi-modal sensor inputs and learning various representations. UniAD pivots to trajectory planning and learns perception and prediction as parallel tasks (Hu et al. 2023). VAD decodes the agent motion and map structure as vectors (Jiang et al. 2023). FlowDrive predicts an interpretable flow field for safer planning (Jiang et al. 2025). Since the BEV space can be heavy, recent works have proposed to generate trajectory plans directly from multi-view camera features, e.g., SparseDrive (Sun et al. 2025) and DrivoR (Kirby et al. 2026).

Driving in the real world involves considerable uncertainty, and thus modern planners often generate and score multimodal predictions. VAD-V2 builds a planning vocabulary with possible trajectories, and predicts a probability distribution over the action space (Chen et al. 2024b). Hydra-MDP learns to score the vocabulary with both human demonstration and rule-based experts (Li et al. 2025). DiffusionDrive (Liao et al. 2025) denoises anchored Gaussian distributions, and GoalFlow (Xing et al. 2025) generates a goal-conditioned plan with Flow Matching.

Synthetic Views in Autonomous Driving

Traffic simulators such as CARLA, MetaDrive and Hugsim are important platforms for training and evaluation (Dosovitskiy et al. 2017; Li et al. 2022; Zhou et al. 2024). From simulation, diverse driving scenarios and rendered sensor observations can be generated, which are essential for safety-critical decisions (Liu et al. 2026). With the recent advancements in neural rendering, such as Gaussian Splatting, more realistic 3D scenes can be built from the real-world driving logs (Kerbl et al. 2023). SimScale samples trajectories from reconstructed 3D scenes and uses them as augmented data for training more robust policies (Tian et al. 2025). RAD trains the driving policy to fully explore the scene via reinforcement learning (RL) (Gao et al. 2026).

Another thread of work has focused on the semantic layout of the scene instead of pursuing photorealism. Xu, Tan, and Kong (2018) and Müller et al. (2018) train a driving policy based on semantic segmentation results of the ego view. Chung et al. (2022) and Behl et al. (2020) propose to ab-

stract away the visual appearance details with instance-level segmentation masks. RAP renders the camera views for the ego agent using 3D bounding boxes of traffic participants, and aligns the feature space distribution of real camera input with the rendered semantic views. As a result, the model can focus on the semantics of the scenes, e.g., lane centerlines, road boundary and surrounding agents, and stay robust to the drifts in visual appearance. Gigapixel similarly renders a simplified bounding-box world to support pixel-space RL-based self-play (Rowe et al. 2026). Such semantic layouts have also been utilized as control images in Cosmos3 for rendering photo-realistic views (Agarwal et al. 2026).

Aerial Traffic Monitoring

Drones are popular platforms for collecting trajectory data due to their high mobility and the low occlusion of aerial perspective. For example, HighD and inD record traffic on highways and at intersections, respectively (Krajewski et al. 2018; Bock et al. 2020). SinD additionally includes traffic signal states (Xu et al. 2022), Interaction emphasizes the joint behavior of vehicles (Zhan et al. 2019), and HeteroD points out the importance of vulnerable road users (Chen et al. 2026). However, these datasets are small-scale (e.g., a single drone and a few hours) and are often scattered across many cities. Meanwhile, the transportation community has initiated large-scale coordinated monitoring with multiple drones over continuous urban regions. The pNEUMA experiment monitors the entire city center of Athens, Greece (Barmounakis and Geroliminis 2020), and Songdo Traffic contains geo-referenced vehicle trajectories for complex intersections in Songdo, South Korea (Fonod et al. 2025). SkyDrive repurposes the Songdo Traffic data from mobility analysis to autonomous driving by treating each tracked car as an ego vehicle and building egocentric scenarios.

SkyDrive: Learn to Drive from the Sky

Overview

Figure 1 presents the overall workflow of *SkyDrive*, which turns aerial traffic monitoring into real driving behavior supervision. In the data collection stage, a swarm of drones is deployed over the city to monitor multiple locations simultaneously. Then, the accurate locations of most vehicles can be continuously tracked for the entire duration of their presence in the bird’s-eye-view videos, encoding the driving behavior of those traffic participants. Therefore, the egocentric dynamics of any agent can be learned by centering the coordinate system on it, which brings two-fold benefits. First, the vehicle driving behaviors can be learned without operating instrumented vehicles for a long time. Second, the data volume can be effectively scaled up while ensuring the quality.

With the trajectory data and map information, we primarily focus on vision-based trajectory planning where a model plans the future ego trajectory using semantic images, ego history and a high-level driving direction. We evaluate the planners in terms of accuracy, regulation compliance and safety. Meanwhile, we also experiment with motion prediction where a model predicts possible future ego motions

with the map and historical trajectories. The remainder of this section will describe the dataset and the tasks in detail.

The SongdoDrive Dataset

We build our dataset with the Songdo Traffic (Fonod et al. 2025) dataset, a traffic monitoring dataset collected by a swarm of drones from the modern city of Songdo in South Korea. The dataset provides high-quality geo-referenced vehicle coordinates along with annotated lane bounding boxes, and has been widely utilized in traffic analysis problems. In this work, we repurpose it for trajectory planning as well as motion prediction. To this end, we have developed a data processing pipeline to extract driving scenes, and we refer to the derived dataset as SongdoDrive.

In Songdo Traffic (Fonod et al. 2025), the trajectory data is collected from 20 complex urban intersections (e.g., labeled as A, B or C). The experiment spans four days and each day has 10 monitoring time slots from morning to evening (e.g., 8:00 – 8:30). Although the virtual cameras can be placed on any vehicle, we focus on the cars with valid movements, i.e., the 50% speed quantile is above 10 km/h. Then, we progressively select instances from the candidate pool, and sample 8-second segments with a stride of 4 seconds. The segments with insufficient length, missing locations or invalid size estimations are discarded. In addition, near-duplicate segments are filtered out, e.g., two cars closely following each other. These selected segments are regarded as ego movements, and a driving scenario is then sliced from the monitoring session, which means a scenario contains the complete view of the intersection during the time span of the ego segment.

Using the trajectories and lane bounding boxes, we also derived various types of map information, including the lane directions, stop lines, drivable area polygons and lane connectors. Two such examples are shown in Figure 2, and more of those preprocessing details are in the supplementary materials.

After the preprocessing, SongdoDrive has 650K scenarios from 137.2 hours of drone monitoring logs, and Table 1 shows the overall statistics and the standard train-test splits. After the initial near-static speed filtering, about 368K car trajectories remain, and then ~ 234 K segments are rejected in subsequent data quality control.

Table 1: Statistics of the standard data split

Split	Sessions	Time (h)	Valid veh.	Filtered	Ego segs.
Train	680	116.23	313075	200440	549929
Test	120	20.97	54731	33910	99629
Overall	800	137.20	367806	234350	649558

Table 2 compares SongdoDrive with other motion datasets. NuScenes, Argoverse2 and Waymo are collected with vehicle platforms, while the others are based on drones. The trajectories from drone platforms are notably longer than those from vehicles, and therefore, more ego segments can be sampled from one trajectory. As a result, SongdoDrive has a number of scenarios comparable to those of Waymo and Argoverse2 while requiring less data collection time. Meanwhile, SongdoDrive is larger in scale than other drone-based

datasets, providing more comprehensive coverage of driving behaviors. Besides, the drones in SongdoDrive are deployed over the same urban region, making it a focused solution for adapting autonomous driving models to a specific city.

Table 2: Comparison with other datasets.

Dataset	# unique tracks	Avg track length	#sce.	Scenario duration	Total time
NuScenes	4.3k	–	50K	8s	5.5h
Argoverse2	13.9m	5.16s	250K	11s	763h
Waymo	7.64m	7.04s	576K	9.1s	574h
Interaction	40k	19.8s	–	–	16.5h
SinD	13.2k	–	–	–	7.02h
HeteroD	64.5k	–	–	–	17.5h
SongdoDrive	646.5k	33.8s	650k	8s	137.2h

Trajectory Planning Task

Following Navsim (Dauner et al. 2024), a trajectory planner uses the first 2 seconds of an 8-second ego segment as input information, and is required to plan the ego trajectory for the next 4s. Both the history and prediction are required at 2Hz, with data and ground truth down-sampled from original 30Hz data. Thus, the model has 4 input frames (including the current one) and predicts the next 8 steps. Since the trajectory data is two-dimensional, we lift the driving scene into 3D space by assuming a flat ground surface and a common-sense per-type vehicle height. Then, a set of virtual cameras is applied to the ego vehicles as shown in Figure 1, and the egocentric views can be rendered by rasterizing the 3D vehicle boxes and the map structures. This rendering process follows RAP (Feng et al. 2025), with camera matrices borrowed from (Caesar et al. 2021). Figure 3 shows an example of rendered images in busy traffic.

In addition to the rendered semantic images (2s history at 2Hz), the model also has access to precise ego history, including location, velocity and heading with respect to the current vehicle pose. Considering an autonomous vehicle should know its navigation target, the short-term trajectory planner also knows a high-level driving direction, i.e., left, right or straight, which is obtained offline from the future trajectories. Depending on the design, a planner may choose not to use historical information and proceed only with the current frame (Kirby et al. 2026).

To evaluate the planner, we use the following metrics to cover accuracy, regulation and safety aspects.

- Average Displacement Error (ADE) is the average distance (in meters) between the planned trajectory and the ground truth over all the future steps.
- Final Displacement Error (FDE) is the distance (in meters) between the planned waypoint and the ground-truth waypoint at the last prediction time step.
- Non-Compliant Trajectory (NCT) measures compliance with respect to the drivable area. We report the percentage of trajectories with any waypoints outside the drivable area polygon, i.e., driving off-road.

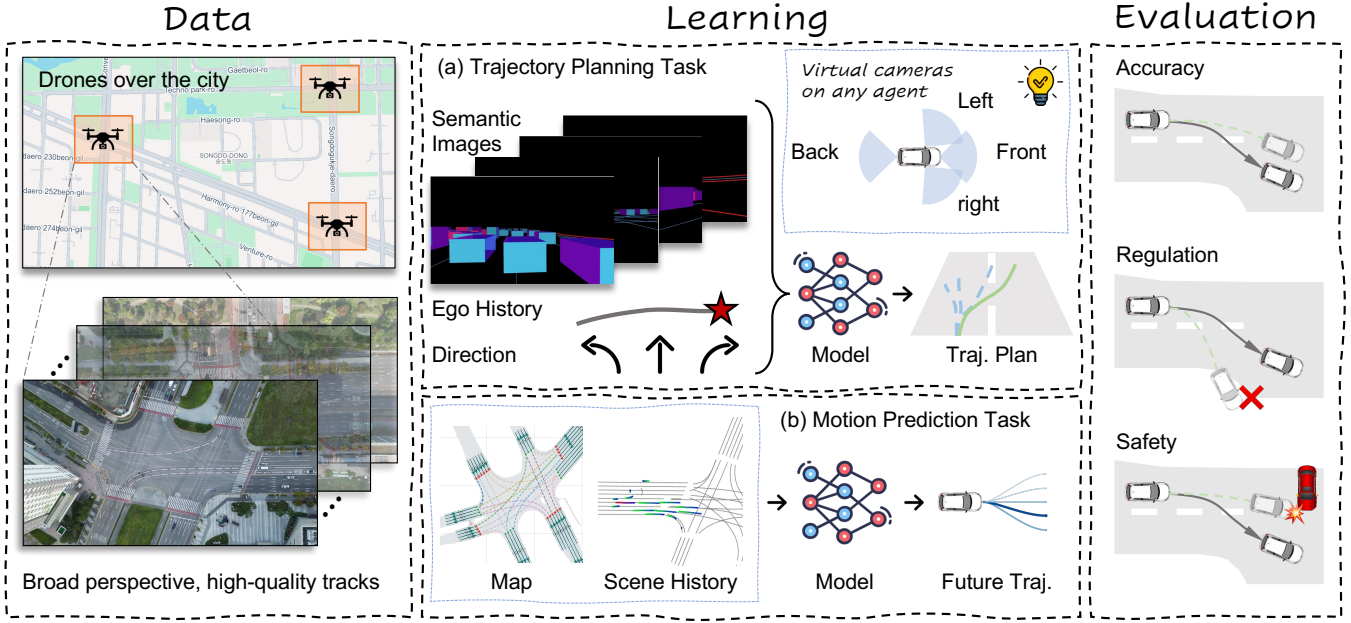


Figure 1: Overview of *SkyDrive*. We utilize accurate vehicle trajectory data from drone-based traffic monitoring research, study trajectory planning and motion prediction tasks, and evaluate accuracy, regulatory compliance and safety.

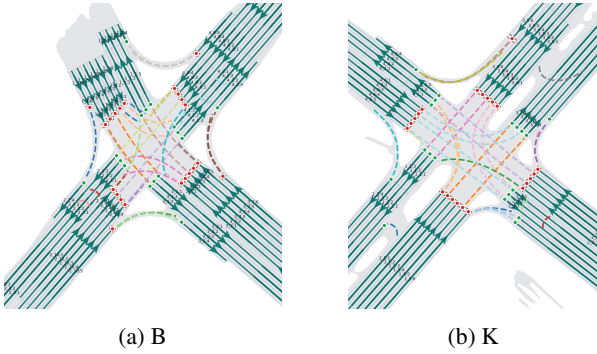


Figure 2: Map visualizations for two example intersections. The lane bounding boxes are labeled in Songdo Traffic, and the stop lines, lane centerlines, and drivable area are derived using trajectory data.

- Time-to-collision (TTC) infraction rate considers the driving risks. At each planned waypoint, we project the ego at a constant velocity and heading 0.3, 0.6 and 0.9 seconds ahead, and check if it collides with any other vehicles. Similar to Navsim (Dauner et al. 2024), the background vehicles will not react to the ego and will proceed according to the logged data. The infraction rate is the percentage of trajectories where the ego has $TTC \leq 0.9s$ at any waypoint.

Since straight moves account for the vast majority of real-world driving, the overall result is likely to be dominated by them. Therefore, we report finer-grained metrics by trajectory types and by Kalman Difficulty (Feng et al. 2024) to evaluate

the performance under typical driving scenarios.

The trajectory types are decided according to the end-point position (4s in the future), as illustrated in Figure 4. All trajectories with displacement less than 3 meters are stationary. Non-stationary trajectories are divided into straight trajectories, (normal) turns and sharp turns according to the angle between the initial heading and the endpoint displacement vector. The turns have left and right directions, with a positive angle indicating left. For example, a trajectory with endpoint (5m, $+75^\circ$) from the start is a sharp left turn. Unlike UniTraj, we do not distinguish U-turns since they can rarely be completed within the shorter prediction horizon.

Kalman Difficulty indicates how much the trajectory differs from naive movements. A constant-velocity Kalman filter is estimated from past trajectories and projected to future steps. The FDE between the Kalman filter prediction and the ground truth is then defined as Kalman Difficulty. The trajectories are then grouped into easy, medium and hard categories based on Kalman Difficulty: < 10 , $[10, 20)$ and ≥ 20 , respectively.

Motion Prediction Task

The motion forecasting task follows UniTraj (Feng et al. 2024). A model has access to map structures, including lanes, stop lines and the drivable area, as well as past trajectories for the ego and surrounding vehicles. Then, the model is required to predict $K = 6$ possible future trajectories of the ego vehicle, along with the probabilities. Concretely, the model receives 2s of history in each 8-second driving scenario and predicts 6s into the future, both at 10Hz. Thus, the input contains 21 frames in total including the current time, and the output requires 60 future steps. Besides, the coordinate

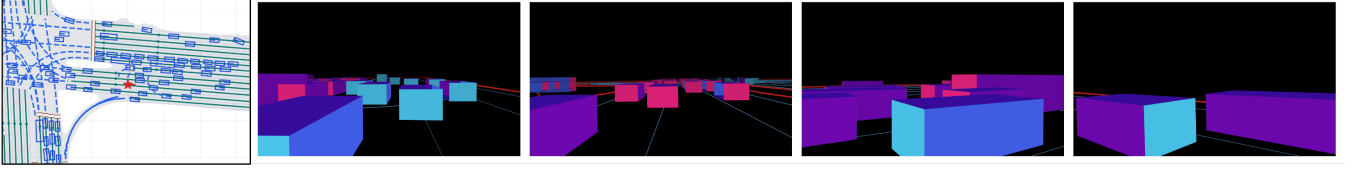


Figure 3: Example of semantic views during a lane-changing event in busy traffic. The leftmost panel shows a bird’s-eye view of the scene, where the ego is rendered as a red star. The perspective views from left to right are front, back, left and right.

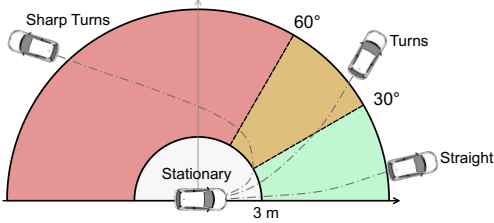


Figure 4: The trajectory types in the trajectory planning task

system is centered on the current ego position while keeping the ego heading pointed to the right. Finally, the predictions are evaluated with:

- minADE and minFDE are the minimum ADE and FDE over the K predicted trajectories.
- BrierFDE is minFDE with a penalty term $(1 - p)^2$, where p is the probability of the best predicted trajectory. This metric encourages the model to assign high confidence to accurate predictions.
- Miss Rate is the percentage of samples where minFDE exceeds a certain threshold (e.g., 2 m).

To summarize, Table 3 compares the input, output, and evaluation of the two tasks on SongdoDrive.

Experiments

Trajectory Planning Settings and Results

We adapt two state-of-the-art methods from NavSim (Dauner et al. 2024) to SongdoDrive, i.e., DrivoR (Kirby et al. 2026) and RAP (Feng et al. 2025). Both models generate multiple trajectory proposals first, and then use a learned scorer to choose the trajectory with the best Predictive Driver Model Score (PDMS). An important difference is that RAP trains the image encoder to align the distributions of real camera images and the synthetic semantic views. In this work, we train them to imitate the ground-truth driving trajectories without any other auxiliary scores for simplicity. For DrivoR, we use the negative of the average of ADE and FDE as the score. For RAP, we use its alternative option based on Rater Feedback Score (Ettinger et al. 2021).

Table 4 shows the overall trajectory planning results using the standard train-test splits. In the zero-shot experiments (Z), the author-released checkpoints of DrivoR and RAP are tested directly on SongdoDrive. In the full data experiments, DrivoR is trained from scratch for 20 epochs and RAP is fine-tuned from the zero-shot checkpoint for 5 epochs since

a full retrain is not possible without paired real and synthetic images. Both models are trained with a batch size of 32 on two H100 GPUs under a similar budget, where DrivoR requires ~ 3 hours/epoch and RAP requires ~ 10 hours/epoch.

When directly applied outside their training domain, both models have unsatisfactory performance. The high ADE, FDE and TTC infraction rate show that the planners deviate from reasonable human choices and frequently pose risks. Although DrivoR’s Navsim performance is very close to RAP’s (93.1 vs 93.7 PDMS), its zero-shot performance on SongdoDrive is less favorable than RAP’s, and this difference results from the image appearance gap. Since RAP was trained to align the rendered semantic views with real camera images, the rendered semantic views are closer to its training domain. In contrast, DrivoR was trained only with real images, and its domain gap is larger than RAP’s. The NCT rate provides a stronger indication. As shown in Figure 3, the semantic view highlights the road boundary in red, and RAP’s regulatory adherence can be better preserved, resulting in a much lower off-road rate of 1.12%.

Compared to the zero-shot experiments, both trained models improve significantly on all metrics, showing the effect of in-domain supervision. The overall FDEs of RAP and DrivoR have been reduced by 66.4% and 57.4%, respectively. The fact that RAP improves more suggests that image-domain alignment can facilitate the learning of city-specific driving behaviors. More exciting progress is observed in safety and regulatory adherence, although they are not explicitly required in training. The TTC infraction rate of DrivoR drops from 38.1% to 10.82% and the metric for RAP decreases from 25.56% to 5.89%, which means both models have learned to navigate more safely and RAP does even better with its aligned image domain knowledge.

To better analyze the performance under different driving scenarios, Figure 5 and Figure 6 show the FDE, TTC and NCT metrics of the four tested models by Kalman Difficulty and trajectory types respectively. Generally, both the trained RAP and DrivoR have better performance in easier scenarios, e.g., the easy category by Kalman Difficulty and the stationary or straight categories by trajectory type. In more challenging scenarios, e.g., sharp turns, the trained models have less favorable performance. In contrast, a clear failure mode for zero-shot methods occurs in the easy cases, e.g., high FDE and TTC on stationary and straight trajectories. This difference suggests that the planners are confused about moving and stopping, and after supervised training with drone data, this distinction can be learned much better.

Table 3: Summary of the trajectory planning and motion prediction tasks.

Task	Input data	Input horizon	Output data	Output horizon	Frequency	Metrics
Trajectory planning	Rendered multi-camera semantic views, ego motion history, high-level direction	2s	Ego trajectory	4s	2Hz	ADE, FDE, TTC infraction rate, NCT
Motion prediction	Map structures and ego/surrounding-vehicle trajectories	2s	6 ego trajectories and probabilities	6s	10Hz	minADE, minFDE, Miss Rate, BrierFDE

Table 4: Overall standard-split trajectory planning results.

Method	Training	ADE (m) ↓	FDE (m) ↓	TTC (%) ↓	NCT (%) ↓
DrivoR	Zero-shot	3.702	8.855	38.10	8.24
DrivoR	Full Data	1.589	3.774	10.82	4.80
RAP	Zero-shot	3.399	8.116	25.56	1.12
RAP	Full Data	1.263	2.730	5.89	0.13

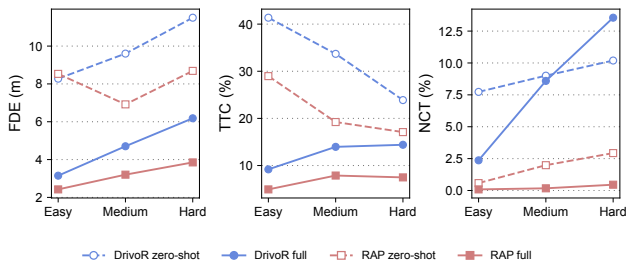


Figure 5: Evaluation results by Kalman difficulty. Easy $\in [0, 10)$, medium $\in [10, 20)$, hard ≥ 20 .

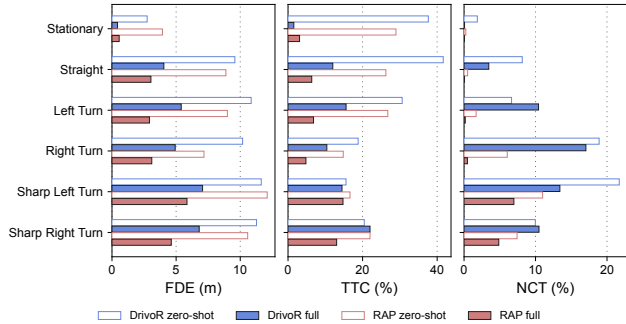


Figure 6: Trajectory planning results by trajectory type

Data Scaling and Cross-Intersection Performance

This section further investigates how much data is needed to have a significant performance gain. To make the data scaling compatible with feasible real-world drone monitoring, we make subsets of the standard training set by monitoring sessions identified by intersection, day and time slot. The sessions are selected via round-robin scheduling over intersections to spread the monitoring efforts evenly over the space. For example, the 1% subset has 6 sessions out of the 680 training sessions, and they belong to 6 random intersections. The model is trained with the same hyperparameters

on the training subset and evaluated on the complete test set.

Figure 7 shows the data scaling curve for the four metrics with the zero-shot performance as a reference. The results show that even 1% data supervision can make a significant difference, with all metrics substantially improved. While RAP consistently improves with more training data, DrivoR’s performance improves much more slowly, which highlights the importance of aligning the image representation space.

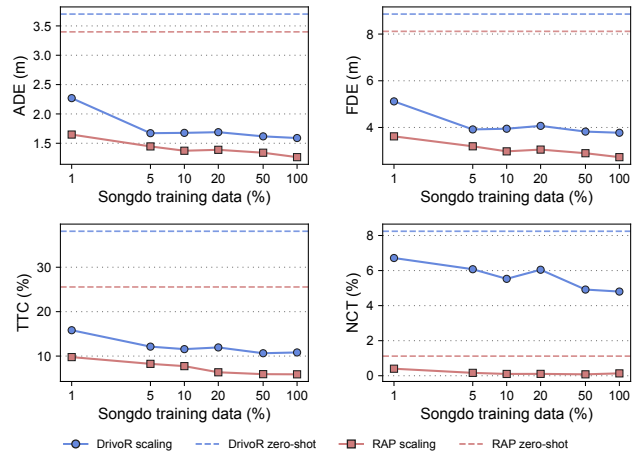


Figure 7: Scaling experiment results. The experiments with 1%, 5%, 10%, 20% and 50% training data have 6, 34, 68, 136 and 340 monitoring sessions respectively, corresponding to 1.0, 5.8, 11.7, 23.3 and 58.3 hours of approximate traffic monitoring time.

In addition to domain gaps across cities and shifts in image appearance, generalization gaps may also exist among different locations in the same city. To provide a quantified analysis, we make a cross-intersection data split, train the models on 17 intersections and test on the remaining three. Meanwhile, the standard train split covers all intersections.

To make a fair comparison, we take the overlapping samples from the test sets of the standard (S) and cross-

intersection (C) splits, and evaluate the models trained with the corresponding train sets. Table 5 shows those evaluation results, and again confirms the similarity of different intersections in the same region. The models trained with the standard split are generally preferable, since they are trained with data from all locations, but the cross-intersection gaps are not significant. The only relatively large performance gap is observed for the NCT of DrivoR, where the cross-intersection model struggles more with the road boundaries.

Table 5: Matched standard (S) and cross-intersection (C) results on overlapping samples

Method	ADE (m) ↓	FDE (m) ↓	TTC (%) ↓	NCT (%) ↓
DrivoR (S)	1.485	3.514	8.44	4.72
DrivoR (C)	1.535	3.659	7.94	7.87
RAP (S)	1.145	2.447	4.07	0.09
RAP (C)	1.170	2.506	5.09	0.06

Motion Prediction Settings and Results

In this section, we benchmark the performance of AutoBot (Girgis et al. 2022), MTR (Shi et al. 2022), and Wayformer (Nayakanti et al. 2023). The implementation is based on UniTraj and default configurations are kept for all methods. Table 6 presents the overall evaluation metrics. MTR* denotes an MTR model trained on nuScenes and evaluated zero-shot on SongdoDrive (Caesar et al. 2020).

Table 6: Motion prediction results. Best values in bold.

Method	BrierFDE	minADE	minFDE	Miss Rate
MTR*	3.953	1.497	3.369	0.494
AutoBot	1.847	0.588	1.197	0.154
Wayformer	1.763	0.564	1.157	0.148
MTR	1.902	0.680	1.516	0.273

Table 7: BrierFDE by Kalman difficulty. Best values in bold.

Method	Easy [0, 30)	Medium [30, 50)	Hard [50, 100]
MTR*	3.248	8.088	16.975
AutoBot	1.722	2.600	3.564
Wayformer	1.645	2.482	3.242
MTR	1.765	2.716	4.107

Among the models trained on SongdoDrive, Wayformer achieves the best performance across all metrics, while AutoBot is a close second. Notably, the MTR model is a larger model with higher per-epoch training time, but it still delivers slightly worse performance than the lightweight AutoBot. Compared to the trained MTR model, the zero-shot MTR* has 108% higher BrierFDE and 120% higher minADE. This severe degradation confirms the pronounced domain shift and the necessity of in-domain supervision.

Table 7 shows the BrierFDE by Kalman difficulty, where the errors of all methods increase from easy to hard cases.

Table 8: BrierFDE by trajectory type. Best values in bold.

Type / Method	MTR*	AutoBot	Wayformer	MTR
Stationary	2.017	0.623	0.562	0.457
Straight	2.704	1.791	1.690	1.817
Straight-right	5.337	2.700	2.631	2.985
Straight-left	4.549	2.423	2.262	2.668
Right turn	7.825	2.459	2.399	2.667
Left turn	7.131	2.396	2.350	2.554
Right U-turn	6.546	6.222	4.857	5.993
Left U-turn	12.252	4.513	4.951	5.024

Wayformer remains the best method across all three difficulty levels, with a more obvious advantage on the hard subset. Meanwhile, compared to zero-shot inference, the trained MTR model achieves a 75.8% improvement on the hard cases. Table 8 shows a per-type breakdown. Consistent with the stop-and-go confusion in Figure 6, MTR* has the largest relative increase in error (+341% compared to trained MTR) for the stationary type. Other than that, the straight type is easier to handle, whereas turnings are more challenging.

Conclusions

In this work, we introduce *SkyDrive* to turn drone-based traffic monitoring data into a supervision source for autonomous driving. Owing to the extended aerial field of view, *SkyDrive* can efficiently obtain diverse and high-quality trajectories from many vehicles, and create more training samples within the same operation time. We quantified the generalization gap when applying trajectory planners and motion predictors directly in an unseen city, and identified confusion between stopping and moving as a common failure mode. The experiments with RAP and DrivoR demonstrated that trajectory planners can be significantly improved with a small amount of data corresponding to ~ 30 minutes of traffic monitoring per location, and also highlighted the importance of training the image encoder to understand the semantic layout of driving scenarios. Our investigation has shown that drone-based traffic monitoring is a favorable choice for adapting autonomous driving models to a new city, and we hope this finding can offer a new perspective for practitioners in the autonomous driving community.

References

- Agarwal, N.; Ali, A.; Allen, J.; Antolini, M.; Aubame, A.; Azzolini, A.; Bai, J.; Bala, M.; Balaji, Y.; Bapst, J.; et al. 2026. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*.
- Barmounakis, E.; and Geroliminis, N. 2020. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Transportation research part C: emerging technologies*, 111: 50–71.
- Behl, A.; Chitta, K.; Prakash, A.; Ohn-Bar, E.; and Geiger, A. 2020. Label efficient visual abstractions for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2338–2345. IEEE.

- Bock, J.; Krajewski, R.; Moers, T.; Runde, S.; Vater, L.; and Eckstein, L. 2020. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1929–1934. IEEE.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.
- Caesar, H.; Kabzan, J.; Tan, K. S.; Fong, W. K.; Wolff, E.; Lang, A.; Fletcher, L.; Beijbom, O.; and Omari, S. 2021. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*.
- Chen, D.; and Krähenbühl, P. 2022. Learning from all vehicles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17222–17231.
- Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2024a. End-to-end Autonomous Driving: Challenges and Frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Chen, S.; Jiang, B.; Gao, H.; Liao, B.; Xu, Q.; Zhang, Q.; Huang, C.; Liu, W.; and Wang, X. 2024b. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*.
- Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3339–3348.
- Chen, Y.-H.; Chang, W.-J.; Kotulla, C.; Keutgens, T.; Runde, S.; Moers, T.; Klas, C.; Zhan, W.; Tomizuka, M.; and Chen, Y.-T. 2026. HetroD: A High-Fidelity Drone Dataset and Benchmark for Autonomous Driving in Heterogeneous Traffic. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Chung, S.-H.; Kong, S.-H.; Cho, S.; and Nahrendra, I. M. A. 2022. Segmented encoding for Sim2Real of RL-based end-to-end autonomous driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, 1290–1296. IEEE.
- Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; Geiger, A.; and Chitta, K. 2024. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.
- Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C. R.; Zhou, Y.; et al. 2021. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9710–9719.
- Feng, L.; Bahari, M.; Amor, K. M. B.; Zablocki, É.; Cord, M.; and Alahi, A. 2024. UniTraj: A Unified Framework for Scalable Vehicle Trajectory Prediction. *arXiv preprint arXiv:2403.15098*.
- Feng, L.; Gao, Y.; Zablocki, E.; Li, Q.; Li, W.; Liu, S.; Cord, M.; and Alahi, A. 2025. RAP: 3D Rasterization Augmented End-to-End Planning. *arXiv:2510.04333*.
- Fonod, R.; Cho, H.; Yeo, H.; and Geroliminis, N. 2025. Advanced computer vision for extracting georeferenced vehicle trajectories from drone imagery. *Transportation Research Part C: Emerging Technologies*, 178: 105205.
- Gao, H.; Chen, S.; Jiang, B.; Liao, B.; Shi, Y.; Guo, X.; Pu, Y.; Li, X.; Liu, W.; Zhang, Q.; et al. 2026. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 32551–32576.
- Girgis, R.; Golemo, F.; Codevilla, F.; Weiss, M.; D’Souza, J.; Ebrahimi Kahou, S.; Heide, F.; and Pal, C. J. 2022. Latent Variable Sequential Set Transformers for Joint Multi-Agent Motion Prediction. In *International Conference on Learning Representations (ICLR)*.
- Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17853–17862.
- Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8350.
- Jiang, H.; Zhang, Z.; Gao, Y.; Sun, Z.; Wang, Y.; Heng, Y.; Wang, S.; Chai, J.; Chen, Z.; Zhao, H.; et al. 2025. Flowdrive: Energy flow field for end-to-end autonomous driving. *arXiv preprint arXiv:2509.14303*.
- Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).
- Kirby, E.; Boulch, A.; Xu, Y.; Yin, Y.; Puy, G.; Zablocki, A.; Bursuc, A.; Gidaris, S.; Marlet, R.; Bartoccioni, F.; et al. 2026. Driving on registers. *arXiv preprint arXiv:2601.05083*.
- Krajewski, R.; Bock, J.; Kloeker, L.; and Eckstein, L. 2018. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st international conference on intelligent transportation systems (ITSC)*, 2118–2125. IEEE.
- Li, K.; Li, Z.; Lan, S.; Xie, Y.; Zhang, Z.; Liu, J.; Wu, Z.; Yu, Z.; and Alvarez, J. M. 2025. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*.
- Li, Q.; Peng, Z.; Feng, L.; Zhang, Q.; Xue, Z.; and Zhou, B. 2022. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3): 3461–3475.

- Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; et al. 2025. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12037–12047.
- Liu, Y.; Peng, Z. M.; Cui, X.; and Zhou, B. 2026. Adv-bmt: Bidirectional motion transformer for safety-critical traffic scenario generation. *Advances in Neural Information Processing Systems*, 38: 55310–55335.
- Müller, M.; Dosovitskiy, A.; Ghanem, B.; and Koltun, V. 2018. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*.
- Naeinian, F.; Hamza, A.; Zhu, H.; and Choromanska, A. 2026. Zero-Shot Cross-City Generalization in End-to-End Autonomous Driving: Self-Supervised versus Supervised Representations. *arXiv preprint arXiv:2603.11417*.
- Nayakanti, N.; Al-Rfou, R.; Zhou, A.; Goel, K.; Refaat, K. S.; and Sapp, B. 2023. Wayformer: Motion Forecasting via Simple & Efficient Attention Networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2980–2987. IEEE.
- Prakash, A.; Chitta, K.; and Geiger, A. 2021. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7077–7087.
- Rowe, L.; Girgis, R.; de Schaetzen, R.; Cornelisse, D.; Grandhi, A.; Heide, F.; Vinitzky, E.; Pal, C.; and Paull, L. 2026. Scaling Self-Play for End-to-End Driving. *arXiv preprint arXiv:2606.19641*.
- Shi, S.; Jiang, L.; Dai, D.; and Schiele, B. 2022. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35: 6531–6543.
- Sun, W.; Lin, X.; Shi, Y.; Zhang, C.; Wu, H.; and Zheng, S. 2025. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 8795–8801. IEEE.
- Tian, H.; Li, T.; Liu, H.; Yang, J.; Qiu, Y.; Li, G.; Wang, J.; Gao, Y.; Zhang, Z.; Wang, L.; et al. 2025. Simscale: Learning to drive via real-world simulation at scale. *arXiv preprint arXiv:2511.23369*.
- Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J. K.; et al. 2023. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*.
- Xing, Z.; Zhang, X.; Hu, Y.; Jiang, B.; He, T.; Zhang, Q.; Long, X.; and Yin, W. 2025. GoalFlow: Goal-Driven Flow Matching for Multimodal Trajectories Generation in End-to-End Autonomous Driving. *arXiv preprint arXiv:2503.05689*.
- Xu, N.; Tan, B.; and Kong, B. 2018. Autonomous driving in reality with reinforcement learning and image translation. *arXiv preprint arXiv:1801.05299*.
- Xu, Y.; Shao, W.; Li, J.; Yang, K.; Wang, W.; Huang, H.; Lv, C.; and Wang, H. 2022. SIND: A drone dataset at signalized intersection in China. *arXiv preprint arXiv:2209.02297*.
- Yang, J.; Shi, S.; Wang, Z.; Li, H.; and Qi, X. 2021. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10368–10378.
- Yao, Y.; Yan, S.; Goehring, D.; Burgard, W.; and Reichardt, J. 2024. Improving out-of-distribution generalization of trajectory prediction for autonomous driving via polynomial representations. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 488–495. IEEE.
- Yasarla, R.; Han, S.; Cheng, H.-P.; Bhattacharyya, A.; Mahajan, S.; Liu, L.; Shi, Y.; Garrepalli, R.; Cai, H.; and Porikli, F. 2025. Roca: Robust cross-domain end-to-end autonomous driving. *arXiv preprint arXiv:2506.10145*.
- Zhan, W.; Sun, L.; Wang, D.; Shi, H.; Clausse, A.; Naumann, M.; Kummerle, J.; Konigshof, H.; Stiller, C.; de La Fortelle, A.; et al. 2019. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*.
- Zhang, J.; and Ohn-Bar, E. 2021. Learning by watching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12711–12721.
- Zhou, H.; Lin, L.; Wang, J.; Lu, Y.; Bai, D.; Liu, B.; Wang, Y.; Geiger, A.; and Liao, Y. 2024. HUGSIM: A Real-Time, Photo-Realistic and Closed-Loop Simulator for Autonomous Driving. *arXiv preprint arXiv:2412.01718*.
- Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; and Loy, C. C. 2023. Domain Generalization: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4396–4415.