Leveraging Probabilistic Modeling for Robust End-to-End Autonomous Driving across Domains

Rajeev Yasarla Shizhong Han Hsin-Pai Cheng Litian Liu
Shweta Mahajan Apratim Bhattacharyya Yunxiao Shi Risheek Garrepalli
Hong Cai Fatih Porikli
Qualcomm AI Research[*]
{ryasarla,shizhan,hsinpaic,litiliu,shwemaha,aprabhat,yunxshi,rgarrepa
hongcai,fporikli}@qti.qualcomm.com

Abstract

End-to-end (E2E) autonomous driving has recently emerged as a new paradigm, offering significant potential. However, few studies have looked into the practical challenge of deployment across domains. In this work, we propose RoCA, a novel framework for Robust Cross-domain E2E Autonomous driving. RoCA formulates the joint probabilistic distribution over the tokens that encode ego and surrounding vehicle information in the E2E pipeline. Instantiating with a Gaussian process (GP), RoCA learns a set of basis tokens with corresponding trajectories, which span diverse driving scenarios. Then, given any driving scene, it is able to probabilistically infer the future trajectory. By using RoCA together with a base E2E model in source-domain training, we improve the generalizability of the base model, without requiring extra inference computation. In addition, RoCA enables robust adaptation on new target domains, significantly outperforming direct finetuning. We extensively evaluate RoCA on various cross-domain scenarios and show that it achieves strong domain generalization and adaptation performance.

1 Introduction

In autonomous driving, most recent research has shifted towards integrated, end-to-end (E2E) systems [2, 3, 4, 8, 11, 16]. While E2E approaches can potentially provide enhanced overall driving performance thanks to the joint optimization across components, their robustness can be lacking when encountering less frequent scenarios. An important factor is the lack of diversity in existing large-scale training datasets *e.g.,* [1, 5, 6], which often fail to capture the full spectrum of driving scenarios. For instance, datasets like nuScenes [1] are dominated by simple events, with limited coverage of rare, safety-critical edge cases. This imbalance is further amplified by standard training protocols, which tend to prioritize performance on frequent scenarios, causing the optimization to under-weigh long-tail events. As a result, E2E models trained in such a ways have sub-optimal performance when deployed in different domains, such as different cities, lighting environments, camera characteristics, or weather conditions.

To address these challenges, we propose RoCA, Robust Cross-domain end-to-end Autonomous driving. RoCA is an end-to-end autonomous driving framework designed for enhanced robustness and efficient adaptation using only multi-view images. RoCA learns a compact yet comprehensive codebook containing basis token embeddings ($\mathbf{b}$) that represent diverse ego and agent states, spanning both source and potentially target data characteristics. Crucially, RoCA leverages this learned codebook within a Gaussian Process (GP) framework. During inference, given a new scene's token embedding, the GP probabilistically predicts future ego waypoints and agent motion trajectories

[*] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

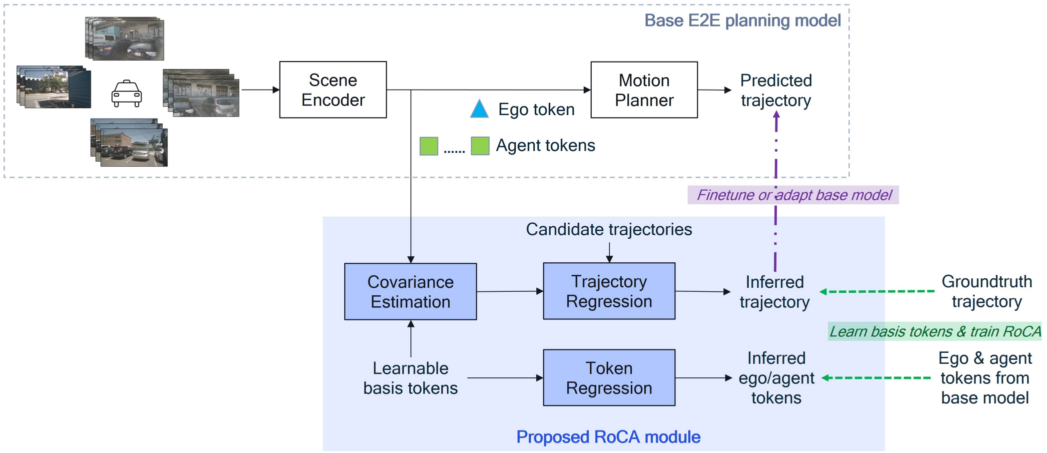

Figure 1: RoCA framework overview.[1] RoCA consists of two components. (1) A base E2E planner (*e.g.,* [11, 14]) extracts the ego and agent tokens from multi-view images for the motion planner to predict future trajectories. (2) Proposed RoCA module, which leverages Gaussian process (GP). In source-domain training, RoCA learns a set of basis tokens from the source domain via reconstructing ego and agent tokens from the basis, supervised by the tokens from the base model (bottom dashed arrow). Its GP-based trajectory regression model predicts trajectories which are supervised by ground-truth waypoints (top dashed arrow). During adaptation, RoCA generates pseudo ground truth to fine-tune the base model on the target domain (purple arrow).

by leveraging the correlation between the current embedding and the learned basis embeddings ($\mathbf{b}$) and their associated known trajectories ($\mathbf{w} = g(\mathbf{b})$) for a learned mapping ($g(.)$) This probabilistic formulation inherently supports generalization, as predictions for novel scenes are informed by their similarity to known embeddings within the diverse codebook. Furthermore, the variance estimated by the GP provides a principled measure of prediction uncertainty. This variance can be used to dynamically weight the training loss, enabling RoCA to automatically assign greater importance to uncertain or difficult predictions. The RoCA framework typically involves an initial stage to build the codebook and optimize GP parameters using source data, followed by efficient deployment or adaptation using only multi-view images processed through the learned GP component.

Our main contributions are summarized as follows:

- We propose RoCA, a novel framework for robust cross-domain end-to-end autonomous driving. Leveraging a Gaussian process (GP) formulation, RoCA captures the joint distribution over ego and agent tokens, which encode their respective future trajectories, enabling probabilistic prediction.
- By utilizing our GP to impose regularization on source-domain training, RoCA leads to more robust end-to-end planning performance across domains.
- RoCA enables adaptation of the end-to-end model on a new target domain. Apart from standard finetuning, its uncertainty awareness makes it possible to select more useful data in the active learning setup.
- Through extensive evaluation, RoCA demonstrates robust performance across domains, for instance, from simulation to real world, across cities. Domain adaptation using RoCA is not only more effective, leading to better planning performance, but also more efficient, by using the predictive uncertainty to select more useful data for finetuning.

2 Proposed Approach: RoCA

We present RoCA, a novel, Gaussian process (GP)-based framework for cross-domain end-to-end autonomous driving. By using a set of basis tokens trained to span diverse driving scenarios, our proposed RoCA module probabilistically infers a trajectory for the current input scene. RoCA not only enhances the robustness of the trained E2E model, but also provides adaptation capability on a new domain.

[1] Images from nuScenes, licensed under CC BY-NC-SA 4.0.

2.1 System Overview

Our proposed E2E pipeline consists of a base E2E model and our proposed RoCA module. The base model, *e.g.,* [11, 14], typically has two parts: 1) a scene encoder $st(.;\theta_{st})$, which converts the input images into scene features/tokens like ego tokens e, and agent tokens a , and 2) a motion planner $h(.;\theta_h)$, which consumes these scene tokens to predict trajectoryego and agent tokens, and predicts the trajectories for both the ego and other vehicles: $\mathrm{p}_{pred}, \mathrm{c}_{pred}, \mathrm{p}_{pred,a}, \mathrm{c}_{pred,a} = h(\mathrm{e}, \mathrm{a}; \theta_h)$, where p denotes the waypoints and c denotes the trajectory class (*e.g.,* total number of classes can be 16 trajectory groups for each driving command of turn left, turn right, and go straight.); θ_{st} and θ_h are learnable parameters. The RoCA module contains a Gaussian process model, denoted by $g(.;\theta_g,\kappa)$, where $\kappa(.)$ is the GP kernel function and θ_g denotes the learnable parameters in th e module.[2] Figure 1 shows the system diagram.

2.2 RoCA Module

2.2.1 Basis Tokens and Trajectories

We construct a "codebook" of learnable basis tokens, $\mathcal{B} = \{\mathbf{B}_k = \{\mathrm{b}_{j,k}\}_{j=1}^{C}\}_{k=1}^{N_{code}}$, where N_{code} is the number of basis groups, each representing a certain trajectory pattern, *e.g.,* turn left, turn right, and C is the group size. These basis tokens are designated to bijectively map to a set of plausible, safe driving trajectories, $\{\mathbf{W}_k\}_{k=1}^{N_{code}}$. To construct this set of basis trajectories, we first sample $N_{code} \cdot C$ representative trajectories from ground-truth human driving data, *e.g.,* nuScenes [1]. They are then clustered into N_{code} groups, such that each group $\mathbf{W}_k$ contains C trajectories with similar driving patterns. In our Gaussian process formulation, each trajectory $\mathrm{w}_{j,k} \in \mathbf{W}_k$ is associated with a unique, learnable basis $\mathrm{b}_{j,k} \in \mathbf{B}_k$. In other words, during training, each basis token learns the driving scenario that corresponds to its trajectory.

2.2.2 Reconstructing Ego and Agent Tokens

Given a driving scenario with ego and agent tokens from the base model, e and a, we first classify them into the respective basis groups. Let $\mathrm{c_e}$ denote the index of the group assigned to e. This classification is performed based on the kernel distance metric and an MLP operating on distance, *i.e.,* MLP($\kappa(\mathrm{e}, \mathbf{B})$) predicts the classification logits for the ego token (similarly for agent).

Let $\mathbf{B}_{\mathrm{c_e}}$ denote the basis tokens in the classified group $\mathrm{c_e}$. The core mechanism for learning the basis $\mathbf{B}_{\mathrm{c_e}}$ is by reconstructing the original ego token e using $\mathbf{B}_{\mathrm{c_e}}$ based on Gaussian process. The joint distribution of e and $\mathbf{B}_{\mathrm{c_e}}$ is given by

$$p(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}}) \sim \mathcal{N}\left(\begin{bmatrix} \mathrm{e} \\ \mathbf{B}_{\mathrm{c_e}} \end{bmatrix}, \begin{bmatrix} \kappa(\mathrm{e}) & \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}}) \\ \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^\top & \kappa(\mathbf{B}_{\mathrm{c_e}}) \end{bmatrix}\right), \tag{1}$$

where $p(.)$ denotes probability density function and $\kappa(.,.)$ is the kernel function evaluating pairwise distances among tokens (specifically, we use the RBF kernel).

The predictive mean $\hat{\mathrm{e}}$ (*i.e.,* the reconstruction of e) and predictive variance σ_{e}^2 are given by

$$\begin{aligned} \hat{\mathrm{e}} &= \mathbf{b}_{anchor,c_e} + \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\bar{\mathbf{B}}_{\mathrm{c_e}}, \\ \sigma_e^2 &= \kappa(\mathrm{e}) - \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^\top + \sigma_{noise}^2\mathbb{I}, \end{aligned} \tag{2}$$

where $\mathbf{b}_{anchor,c_e}$ is the mean of the tokens in group $\mathrm{c_e}$, $\bar{\mathbf{B}}_{\mathrm{c_e}}$ is the zero-mean version of $\mathbf{B}_{\mathrm{c_e}}$, and σ_{noise}^2 is a small, learnable noise variance.

This prediction $\hat{\mathrm{e}}$ serves as an approximation of the original e, reconstructed with the basis tokens. We supervise this reconstruction with the original ego token. Similarly, we applying the same reconstruction process to obtain $\hat{\mathrm{a}}$ and σ_{a} for each agent token a, using their respective classified group of basis tokens $\mathbf{B}_{\mathrm{c_a}}$. The overall reconstruction loss for training the basis tokens is given by

$$\mathcal{L}_{rec} = \frac{1}{\sigma_{\mathrm{e}}^2}|\hat{\mathrm{e}} - \mathrm{e}|^2 - \log(\sigma_{\mathrm{e}}) + \frac{1}{\sigma_{\mathrm{a}}^2}|\hat{\mathrm{a}} - \mathrm{a}|^2 - \log(\sigma_{\mathrm{a}}) + ||\mathbf{B}_{\mathrm{c_a}}\mathbf{B}_{\mathrm{c_a}}^\top - \mathbb{I}||^2 + ||\mathbf{B}_{\mathrm{c_e}}\mathbf{B}_{\mathrm{c_e}}^\top - \mathbb{I}||^2, \tag{3}$$

[2] See [13] for more details on Gaussian processes.

2.2.3 Trajectory Prediction via Gaussian Process

Similar to the previous part, given the ego and agent tokens from the base model, e and a, we first classify them to their respective basis groups, $\mathrm{c_e}$ and $\mathrm{c_a}$. A GP-based regression then infers the future trajectory based on the correlation between the ego/agent token and the basis tokens. The predicted mean and variance for the ego trajectory, $\mathrm{\hat{p}_e}$ and σ_e, is given by

$$\begin{aligned}\mathrm{\hat{p}_w} &= \mathbf{w}_{anchor,\mathrm{c_e}} + \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\bar{\mathbf{W}}_{\mathrm{c_e}},\\ \sigma_w^2 &= \kappa(\mathrm{e}) - \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_{e,w}}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^\top + \sigma^2_{noise}\mathbb{I},\end{aligned} \tag{4}$$

where $\mathbf{w}_{anchor,c_e}$ is the mean of the trajectories in group $\mathrm{c_e}$, $\bar{\mathbf{W}}_{\mathrm{c_e}}$ is the zero-mean version of $\mathbf{W}_{\mathrm{c_e}}$, and σ^2_{noise} is a small, learnable noise variance. The predicted agent trajectory $\mathrm{\hat{p}}_{w,a}$ and variance $\sigma^2_{w,a}$ can be obtained in a similar way.

When training in the source domain, we supervise these GP-based trajectory predictions with the ground truth, as follows:

$$\begin{aligned}\mathcal{L}_{sup} =& \frac{1}{\sigma_\mathrm{w}^2}\mathcal{L}_{planning}(\mathrm{\hat{p}_w}, \mathrm{p}_{gt}) - \log(\sigma_\mathrm{w}) - \frac{1}{\sigma_\mathrm{w,a}^2}\mathcal{L}_{motion}(\mathrm{\hat{p}_{w,a}}, \mathrm{p}_{gt,a}) - \log(\sigma_\mathrm{w,a})\\ &+ \mathcal{L}_{class}(\mathrm{c_e}, \mathrm{c}_{gt,e}) + \mathcal{L}_{class}(\mathrm{c_a}, \mathrm{c}_{gt,a}) + \mathcal{L}_{triplet}(\mathrm{c_e}, \mathrm{c_p}, \mathrm{c_n}) + \mathcal{L}_{triplet}(\mathrm{c_a}, \mathrm{c_{p,a}}, \mathrm{c_{n,a}})\end{aligned} \tag{5}$$

where p_{gt} and $\mathrm{p}_{gt,a}$ are the ground-truth ego and agent trajectories, c_{gt} and $\mathrm{c_{gt,a}}$ are ground-truth ego and agent token categories. The predictive trajectory mean and variance are supervised using variance-weighted losses, similar to those used in [11, 14]). $\mathcal{L}_{planning}$ and $\mathcal{L}_{motion}$ denote the waypoint planning and motion tracking losses, as used in [14, 11].

2.3 Training and Adaptation

2.3.1 Training in Source Domain

Pre-training base E2E model. First, we train the base E2E model on the source domain data following standard training procedure, *e.g.,* [11, 14].

Learning basis tokens and GP parameters. Secondly, we use both $\mathcal{L}_{rec}$ of Eq. 3 and $\mathcal{L}_{sup}$ of Eq. 5 to train RoCA. This includes training the basis tokens and other parameters, *e.g.,* MLP parameters, kernel parameters.

Finetuning base E2E model. Finally, given the trained RoCA module, we utilize it to perform regularized finetuning. More specifically, in addition to the standard supervised loss used to train the base model, we additionally use the following loss by treating RoCA as a teacher:

$$\begin{aligned}\mathcal{L}_{gp} =& \mathcal{L}_{class}(\mathrm{c}_{pred}, \mathrm{c_e}) + \mathcal{L}_{class}(\mathrm{c}_{pred,a}, \mathrm{c_a}) + \mathcal{L}_{triplet}(\mathrm{c}_{pred}, \mathrm{c_p}, \mathrm{c_n}) + \mathcal{L}_{triplet}(\mathrm{c_a}, \mathrm{c_{p,a}}, \mathrm{c_{n,a}})\\ &+ \frac{1}{\sigma_\mathrm{w}^2}\mathcal{L}_{planning}(\mathrm{p}_{pred}, \mathrm{\hat{p}_w}) - \log(\sigma_\mathrm{w}) + \frac{1}{\sigma_\mathrm{w,a}^2}\mathcal{L}_{motion}(\mathrm{p}_{pred,a}, \mathrm{\hat{p}_{w,a}}) - \log(\sigma_\mathrm{w,a})\\ &+ D_{KL}(\mathrm{c}_{pred,e}||\mathrm{c_e}) + D_{KL}(\mathrm{c}_{pred,a}||\mathrm{c_a}),\end{aligned} \tag{6}$$

where p_{pred}, c_{pred}, $\mathrm{c}_{pred,a}$, and $\mathrm{c}_{pred,a}$ are the predicted ego and agent trajectory waypoints and classes from the base E2E model, D_{KL} is the KL-divergence.

2.3.2 Adaptation in Target Domain

In some cases, ground-truth waypoints are available in the target domain, *e.g.,* based on ego status tracking. In such cases, model adaptation is then the same as the final step in source-domain training, where the standard ground-truth supervision on planning is used together with the GP-based regularization from RoCA: $\mathcal{L}_{gp}$ in Eq. 6.

There are scenarios where ground-truth trajectories are not available. For instance, it is nontrivial to process large-volume driving logs and thus, ground-truth waypoints may not be available right after data is collected in the target domain (while images are usually readily available). For unsupervised domain adaptation, as ground-truth labels are not available, we use $\mathcal{L}_{gp}$ to update the base E2E model.

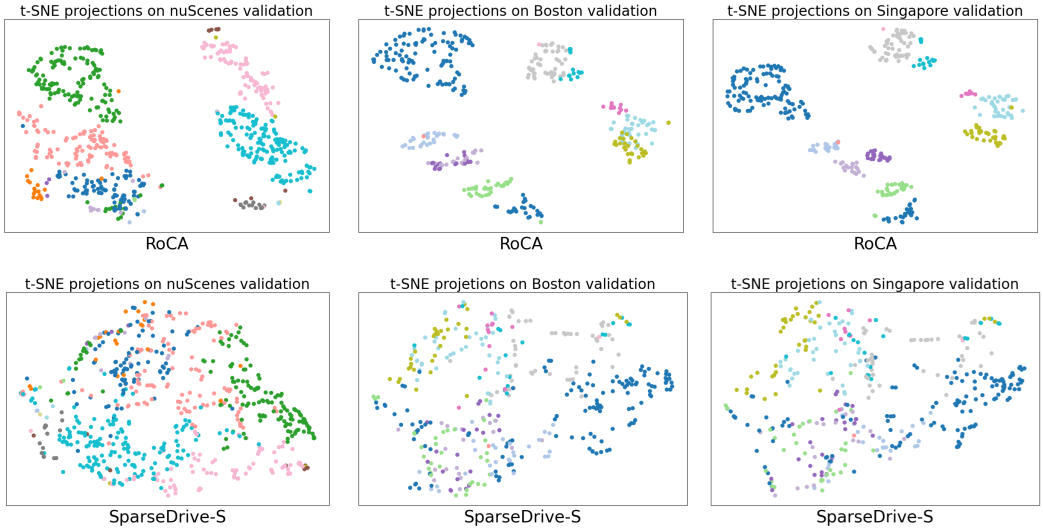

Figure 2: tSNE projection of ego/agent tokens with (top) and without (bottom) RoCA. By using our proposed approach, the model has better separability of different trajectory modes (indicated by different colors). In contrast, the baseline SparseDrive shows poor separability, indicating a sensitivity to any perturbations. The analysis is performed on the full nuScenes val set, the Boston val split, and Singapore val split, with models trained on nuScenes full training set, the Singapore training subset, and Boston training subset, respectively (left, middle, right).

3 Experiments

We conduct extensive experiments to evaluate RoCA on standard end-to-end driving benchmarks and compare it with the latest state-of-the-art (SOTA) methods. Specifically, we consider challenging cross-domain setups to evaluate the robustness and adaptation performance of our proposed approach.

3.1 Datasets, Evaluation Metrics, and Baselines

We use the recent, challenging benchmark of Bench2Drive (B2D) [9], which leverages the CARLA simulator and features 44 difficult interactive scenarios (see Table 1 in B2D paper for more details) across diverse weather and urban conditions. We utilize the B2D base version (1000 video clips) for training and Dev10 for closed-loop evaluation, which is a representative subset selected by the authors [10]. We use the metrics provided by B2D for assessing model performance, e.g., driving score, efficiency.

We use nuScenes [1] to evaluate open-loop planning performance. This dataset consists of 28k total samples in a 22k/6k training/validation split. The objects in each scene are annotated with 3D bounding box, orientation, and speed. nuScenes contains data collected from two cities, Boston and Singapore, which allows us to evaluate cross-domain performance across cities. Within the validation set, we also consider a “targeted" subset containing 689 samples where the vehicle must make a turn, as established in [15]. On nuScenes, we use average L2 trajectory error and average collision rate to evaluate planning performance.

We consider three recent, representative methods, VAD [11], SparseDrive [14], and SSR [12] as the base E2E models. Specifically, we use VAD-T (tiny configuration) and SparseDrive-S (small configuration). Note that our proposed RoCA can be used with any E2E planning model, as long as it provides a tokenized representation.

3.2 Learned Tokens in GP

Our proposed GP-based formulation in RoCA provides more robust scene representation and trajectory planning. Specifically, as shown in Figure 2 (top), the learned basis tokens form clearly distinct clusters (as marked by the different colors), with each cluster representing a different trajectory

Table 1: Closed-loop evaluation on Bench2Drive. Driving scores and efficiency are higher the better, and average L2 error is lower the better.

Method	Driving score	Efficiency	Avg L2
VAD-T	33.75	128.2	1.18
DiMA (Vicuna-v1.5-7B)	36.12	134.2	0.91
SSR	40.36	91.7	0.80
RoCA (VAD-T)	38.57	138.0	0.84
RoCA (SSR)	44.19	100.1	0.68

Table 2: Sim-to-real transfer from Bench2Drive to nuScenes. Driving score and efficiency are higher the better, and average L2 error is lower the better.

Method	**Source: Bench2Drive, Target: nuScenes**							
	Zero-shot				**Fine-tuned**			
	Full Val		Targeted Val		Full Val		Targeted Val	
	Avg L2	Avg Col	Avg L2	Avg Col	Avg L2	Avg Col	Avg L2	Avg Col
VAD-T	1.32	0.51	1.59	0.54	1.01	0.45	1.40	0.46
DiMA (Vicuna-v1.5-7B)	0.94	0.26	1.29	0.38	0.71	0.19	1.06	0.30
SSR	1.08	0.31	1.47	0.44	0.82	0.23	1.30	0.37
RoCA (VAD-T)	0.85	0.24	1.19	0.34	0.63 (0.77)	0.16 (0.20)	0.88 (0.95)	0.26 (0.29)
RoCA (SSR)	0.79	0.22	1.10	0.34	0.57 (0.66)	0.12 (0.17)	0.76 (0.89)	0.25 (0.29)

pattern. Note that the tokens visualized on Boston (Singapore) val subset are from model trained on Singapore (Boston) training subset. As such, even in a new domain, RoCA can still robustly parse the driving scenario in its probabilistic framework and infer a proper trajectory. In contrast, SparseDrive in Figure 2 (bottom) produces mixed pattern across trajectories types, making it fragile when model is deployed in a new domain. For instance, given a token that corresponds to turning left, a slight perturbation to this token (*e.g.,* due to different camera characteristics, lighting, etc.) can result in drastically different driving behavior in the output of the planner.

3.3 In-Domain Evaluation

We perform closed-loop evaluation of models trained on B2D on Dev10, which consists of 10 test scenarios carefully selected by the B2D authors to be both difficult and representative. We compare our proposed RoCA with VAD, SSR [12], and DiMA [7], All methods are trained for 12 epochs. In this evaluation, no adaptation is performed on Dev10. In can be seen from Table 1 that by using our proposed RoCA for training, we achieve significantly better planning performance, even in this in-domain setting. For instance, in the case of VAD-T and SSR, RoCA increases the driving scores by about four points.

3.4 Domain Generalization and Adaptation

We conduct a sim-to-real experiment by transferring the B2D-trained models to nuScenes, either in a zero-shot manner or with a short finetuning of 6 epochs. Specifically, we also evaluate on the more complex targeted split of nuScenes. In Table 2, we see that RoCA has significantly better sim-to-real performance as compared to the baselines as well as a state-of-the-art LLM-based model, in both zero-shot and short-finetuning settings. Moreover, even when we finetune the model without ground truth (only using $\mathcal{L}_{gp}$ of Eq. 6), RoCA still achieves strong performance (in parentheses) when comparing with existing models finetuned with ground truth.

3.5 Active Learning

In the target domain, our goal is to identify the most informative samples for domain adaptation through active learning, thereby reducing both annotation and adaptation costs. To achieve this, we propose using the GP-based predictive variance as a sampling criterion, selecting samples with the highest uncertainty. We compare our variance-based selection with the baseline method of random sampling, evaluated at 5%, 10%, and 15% sampling rates of the full target training data.

Table 3 reports cross-city planning accuracy results for transfers between Singapore and Boston after fine-tuning with ground-truth supervision, using SparseDrive-S as the baseline. Across all sampling

Table 3: Cross-city active learning performance, using 5%, 10%, and 15% target training samples selected randomly or based on predictive variance by RoCA. The base model is SparseDrive-S in this case.

Adapt. Method	Sampling	5%		10%		15%	
		Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓
Singapore → Boston							
Direct finetune	random	0.767	0.215	0.753	0.199	0.711	0.175
Direct finetune	RoCA	0.745	0.191	0.719	0.183	0.678	0.126
RoCA	random	0.644	0.123	0.584	0.121	0.552	0.121
RoCA	RoCA	0.617	0.110	0.554	0.110	0.513	0.108
Boston → Singapore							
Direct finetune	random	0.891	0.198	0.839	0.201	0.823	0.185
Direct finetune	RoCA	0.828	0.192	0.815	0.172	0.793	0.166
RoCA	random	0.707	0.148	0.656	0.133	0.633	0.126
RoCA	RoCA	0.673	0.135	0.604	0.113	0.561	0.102

rates, selection based on RoCA consistently results in lower trajectory errors and collision rates compared to random selection, on both the full validation and the targeted subsets. These results demonstrate the effectiveness of our uncertainty-guided sampling in identifying representative samples for efficient domain adaptation. Furthermore, RoCA consistently outperforms the baseline under both sampling strategies, underscoring its robustness and adaptability in cross-domain scenarios.

4 Conclusions and Discussions

We present RoCA, a novel framework for robust cross-domain end-to-end autonomous driving. By leveraging a GP formulation, RoCA models the joint distribution over ego and agent trajectories, enabling probabilistic prediction and uncertainty-aware planning. This GP-based regularization enhances source-domain training and significantly improves generalization to unseen domains. RoCA's key strength lies in its flexible domain adaptation: it supports standard finetuning, uncertainty-guided active learning, and online adaptation, making it well-suited for real-world deployment. Extensive experiments on Bench2Drive and nuScenes benchmarks show that RoCA provides strong domain generalization and adaptation performance for end-to-end autonomous driving.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[2] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021.

[3] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022.

[4] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15793–15803, 2021.

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

[6] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[7] Deepti Hegde, Rajeev Yasarla, Hong Cai, Shizhong Han, Apratim Bhattacharyya, Shweta Mahajan, Litian Liu, Risheek Garrepalli, Vishal M Patel, and Fatih Porikli. Distilling multi-modal large language models for autonomous driving. *arXiv preprint arXiv:2501.09757*, 2025.

[8] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.

[9] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

[10] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025.

[11] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[12] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025.

[13] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

[14] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

[15] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[16] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.