RoCA: Robust Cross-Domain End-to-End Autonomous Driving

Rajeev Yasarla Shizhong Han Hsin-Pai Cheng Apratim Bhattacharyya Shweta Mahajan† Hong Cai Fatih Porikli

Qualcomm AI Research**

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 (e.g., cities). Although several works have incorporated Large Language Models (LLMs) to leverage their open-world knowledge, LLMs do not guarantee cross-domain driving performance and may incur prohibitive retraining costs during domain adaptation. In this paper, 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

Moving beyond the traditional modular design, where distinct components like perception [24, 30, 42, 58], motion prediction [3, 27, 29], and planning [35] are often developed and optimized in isolation, the focus in autonomous driving research has recently shifted towards integrated, end-to-end (E2E) systems [2, 4, 5, 13, 20, 46]. 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, 8, 9], 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

** Qualcomm AI Research, an initiative of Qualcomm Technologies, Inc. † Work was completed while employed at Qualcomm AI Research

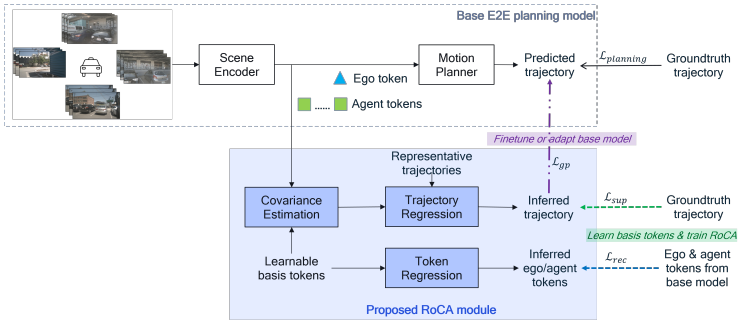

Fig. 1: RoCA framework overview.[1] RoCA consists of two components. (1) A base E2E planner extracts the ego and agent tokens from multi-view images for the motion planner to predict future trajectories. (2) Proposed RoCA module, which leverages Gaussian process (GP). In source-domain training, RoCA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

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 way have sub-optimal performance when deployed in different domains, such as different cities, lighting environments, camera characteristics, or weather conditions.

Large language models (LLMs) have recently emerged as a potentially powerful avenue to address these challenges, as their extensive world knowledge, gleaned from vast internet-scale data, may enable more effective generalization to unseen or rare scenarios [43]. Multi-modal variants (MLLMs) further enhance this by integrating visual inputs [22, 26], enabling a new wave of E2E driving systems that aim to achieve higher interpretability and improved long-tail robustness [12, 16, 34, 37–40]. However, this integration introduces its own set of obstacles. One concern is that there is no inherent guarantee that these LLM-infused models will generalize across disparate domains without further adaptation. Moreover, retraining these massive models for new domains is often prohibitively expensive, demanding large quantities of specialized instruction-following data. Critically, even with broad world knowledge, the fundamental issue of data imbalance may persist, limiting model reliability in safety-critical long-tail situations if not explicitly addressed during training.

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. Instead of relying on pre-trained LLMs, RoCA learns a compact yet comprehensive codebook of basis token embeddings ($\mathbf{b}$) that represent diverse ego states (pose, velocity, acceleration) and agent states (motion trajectory, velocity, acceleration), 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 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, thereby effectively addressing the training imbalance towards common scenarios and improving performance on critical long-tail events.

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. This architecture also naturally lends itself to extensions for online streaming adaptation and active learning.

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 both in domain and 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. Furthermore, RoCA also supports online adaptation.
- Through extensive evaluations (including both closed-loop and long-tail scenarios), RoCA consistently demonstrates robust performance across domains, for instance, transferring from simulation to real world, driving in different cities, and under challenging image degradations. Moreover, domain adaptation with RoCA is not only more effective yielding superior planning accuracy, but is also more efficient, as it leverages predictive uncertainty to prioritize the most informative data for fine-tuning.

2 Related Work

Autonomous driving (AD) systems have evolved from modular pipelines [3,6,24] to end-to-end (E2E) architectures mapping sensor inputs to trajectories [2,5]. Vision-only E2E models such as UniAD [14], VAD [20], SSR [23], and SparseDrive [36] improve efficiency but struggle with domain shifts and rare long-tail events [9]. Recent work addresses these challenges via graph-based transformers [28], domain-invariant objectives [15, 41], and normalization techniques [31, 57], yet robust

handling of safety-critical edge cases remains difficult. Gaussian Processes (GPs) provide principled uncertainty modeling [45, 52–56] and have aided domain adaptation in related fields [11, 21, 25, 50, 51], but their integration into E2E AD is underexplored. This work introduces a GP-based module to enhance trajectory prediction and improve generalization to long-tail scenarios.

3 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, 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.

3.1 Problem Formulation and Architecture

Consider a driving scene at time t consisting of an ego vehicle and a set of surrounding agents. Given a sequence of multi-view images I_t, along with the ego status information (that includes pose, velocity, acceleration), the objective of RoCA is to jointly predict: (i) an ego trajectory $(\mathrm{p_w}, \mathrm{c_w})$ that supports safe and efficient driving, and (ii) the motion trajectories of surrounding agents $(\mathrm{p_{w,a}}, \mathrm{c_{w,a}})$. Our proposed end-to-end (E2E) pipeline consists of two components: a base E2E model and the RoCA module. The base model, *e.g.* [20, 36], typically includes: 1) a scene encoder $st(\cdot; \theta_{st})$, which converts input images into scene features/tokens, and 2) a motion planner $h(\cdot; \theta_h)$, which consumes these tokens to predict trajectories; θ_{st} and θ_h are learnable parameters. Figure 1 illustrates the overall system architecture.

RoCA Preliminaries of GP. To enhance robustness across domains, RoCA incorporates a Gaussian Process (GP) model. A GP can be viewed as an infinite collection of random variables such that any finite subset follows a joint Gaussian distribution. For a set of inputs $V = \{v_1, \ldots, v_n\}$, the corresponding function evaluations satisfy:

$$f(V) = [f(v_1), \ldots, f(v_n)]^\top \sim \mathcal{N}(\mu,\, K(V,V) + \sigma_\epsilon^2 I), \tag{1}$$

where μ denotes the mean vector and $K(V,V)$ is the covariance matrix induced by the kernel function. Predictions at unlabeled points are obtained by conditioning on the observed data, yielding a closed-form Gaussian posterior. The RoCA module instantiates this GP-based formulation as $g(\cdot; \theta_g, \kappa)$, where $\kappa(\cdot)$ is the kernel function and θ_g represents the learnable parameters.[1]

Base E2E model. The scene encoder extracts features from the input multi-camera images and cross-attends learnable queries/tokens with these features. Specifically, the scene encoder produces ego tokens e, and agent tokens a (among other possible tokens), which encode key information for the ego vehicle and

[1] See [33] for additional details on Gaussian Processes.

of the other surrounding vehicles. The motion planner then takes the ego 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.)

RoCA module. The GP module contains learned basis tokens, as well as a set of candidate trajectories that one-to-one correspond to the basis tokens. The GP contains learned basis tokens, and for each of the token, there is a matching representative trajectory. By computing the correlation between e and a and the basis tokens via the kernel function κ, the GP conditionally infers the future ego and agent trajectories, $\mathrm{p}_{\mathrm{w}}, \mathrm{c}_{\mathrm{w}}, \mathrm{p}_{\mathrm{w,a}}, \mathrm{c}_{\mathrm{w,a}} = g_{ego}(\mathrm{e}, \mathrm{a}; \theta_g, \kappa)$.

Following [36], we use an anchor-based method to predict trajectories in both the base motion planner and RoCA module. More specifically, the model first classifies the future trajectory into one of the predefined groups: N_{ego} groups for the ego car and N_{agent} groups for other cars, and then, predicts a residual. The final predicted trajectory is obtained by adding the classified anchor trajectory and the predicted residual.

3.2 RoCA Module

In this part, we discuss our proposed RoCA module in details. This module allows to create an informed and diverse codebook of the plausible trajectories based on prior or pre-existing scenarios. The key advantage is that it can effectively infer trajectories under uncertainty or in new domains by performing a similarity "lookup" to the basis in the codebook.

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. Each basis token is a D-dimensional vector $b_{j,k} \in \mathbb{R}^D$. These basis tokens should learn to span the space of ego and agent tokens, e and a, encountered across diverse driving scenes.

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 train 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. With these basis tokens and trajectories, given a new driving scenario encoded by e and a, we can infer the ego and agent trajectories based on the correlation between the ego/agent tokens and the basis tokens. Within the N_{code} groups, we designate N_{ego} groups to represent distinct ego-car waypoint patterns and N_{agent} groups for various types of agent trajectories.

Reconstructing Ego and Agent Tokens The basis tokens $\mathcal{B} = \{\mathbf{B}_k = \{\mathrm{b}_{j,k}\}_{j=1}^{C}\}_{k=1}^{N_{code}}$ should capture the manifold of ego and agent tokens across diverse driving scenarios. In order to train them, we derive the first set of losses via reconstruction of the original ego and agent tokens given by the base model from the basis 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. Specifically, the ego token is classified into one of the N_{ego} groups and agent token into one of the N_{agent} 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.* $\mathrm{MLP}(\kappa(\mathrm{e}, \mathbf{B}))$ 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

$$p(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}}) \sim \mathcal{N}\left(\begin{bmatrix} \mathrm{e} \\ \mathbf{B}_{\mathrm{c_e}} \end{bmatrix}, \begin{bmatrix} \kappa(\mathrm{e}) & \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}}) \\ \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^\top & \kappa(\mathbf{B}_{\mathrm{c_e}}) \end{bmatrix}\right), \tag{2}$$

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

$$\begin{aligned} \hat{\mathrm{e}} &= \mathbf{b}_{anchor,c_e} + \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\bar{\mathbf{B}}_{\mathrm{c_e}}, \\ \sigma_e^2 &= \kappa(\mathrm{e}) - \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^\top + \sigma_{noise}^2\mathbb{I}, \end{aligned} \tag{3}$$

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

$$\begin{aligned} \mathcal{L}_{rec} = &\frac{1}{\sigma_{\mathrm{e}}^2}|\hat{\mathrm{e}} - \mathrm{e}|^2 + \log(\sigma_{\mathrm{e}}) + \frac{1}{\sigma_{\mathrm{a}}^2}|\hat{\mathrm{a}} - \mathrm{a}|^2 + \log(\sigma_{\mathrm{a}}) \\ &+ ||\mathbf{B}_{\mathrm{c_a}}\mathbf{B}_{\mathrm{c_a}}^\top - \mathbb{I}||^2 + ||\mathbf{B}_{\mathrm{c_e}}\mathbf{B}_{\mathrm{c_e}}^\top - \mathbb{I}||^2, \end{aligned} \tag{4}$$

where the first four terms are based on maximum likelihood under Gaussian assumption and the last two terms encourages orthogonality of the basis. When learning the basis tokens and the parameters in the GP, we treat the original ego and agent tokens e and a as fixed targets, *i.e.* no gradients flow through them.

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, $\hat{\mathrm{p}}_{\mathrm{e}}$ and σ_{e}, is given by

$$\begin{aligned} \hat{\mathrm{p}}_{\mathrm{w}} &= \mathbf{w}_{anchor,\mathrm{c_e}} + \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\bar{\mathbf{W}}_{\mathrm{c_e}}, \\ \sigma_w^2 &= \kappa(\mathrm{e}) - \kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})\kappa(\mathbf{B}_{\mathrm{c_e}})^{-1}\kappa(\mathrm{e}, \mathbf{B}_{\mathrm{c_e}})^{\top} + \sigma_{noise}^2\mathbb{I}, \end{aligned} \tag{5}$$

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_{noise}^2 is a small, learnable noise variance. The predicted agent trajectory $\hat{\mathrm{p}}_{w,a}$ and variance $\sigma_{w,a}^2$ can be obtained similarly.

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

$$\begin{aligned} \mathcal{L}_{sup} = &\frac{1}{\sigma_{\mathrm{w}}^2}\mathcal{L}_{plan}(\hat{\mathrm{p}}_{\mathrm{w}}, \mathrm{p}_{gt}) + \log(\sigma_{\mathrm{w}}) + \mathcal{L}_{cls}(\mathrm{c_e}, \mathrm{c}_{gt,e}) \\ &+ \frac{1}{\sigma_{\mathrm{w,a}}^2}\mathcal{L}_{mot}(\hat{\mathrm{p}}_{\mathrm{w,a}}, \mathrm{p}_{gt,a}) + \log(\sigma_{\mathrm{w,a}}) + \mathcal{L}_{cls}(\mathrm{c_a}, \mathrm{c}_{gt,a}) \\ &+ \mathcal{L}_{tpt}(\mathrm{c_e}, \mathrm{c_p}, \mathrm{c_n}) + \mathcal{L}_{tpt}(\mathrm{c_a}, \mathrm{c_{p,a}}, \mathrm{c_{n,a}}) \end{aligned} \tag{6}$$

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 [20, 36]). $\mathcal{L}_{plan}$ and $\mathcal{L}_{mot}$ denote the waypoint planning and motion tracking losses, as used in [20, 36]. To further refine the embedding space, we utilize triplet loss [32]. For each ego/agent class prediction, we identify three positive classes ($\mathrm{c_p}$), which exhibit similar driving patterns as the ground-truth class (*e.g.* turn left with slightly different angles), and three negative classes ($\mathrm{c_n}$), which have driving behaviors different from the true class (*e.g.* turn left vs. turn right). The triplet loss encourages greater separation between distinct driving categories, and tighter clustering among the same class or similar classes.

3.3 Training and Adaptation

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.* [20, 23, 36].

Learning basis tokens and GP parameters. Secondly, we use both $\mathcal{L}_{rec}$ of Eq. 4 and $\mathcal{L}_{sup}$ of Eq. 6 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:

$$\begin{aligned}\mathcal{L}_{gp} = \mathcal{L}_{cls}(\mathrm{c}_{pred}, \mathrm{c}_{\mathrm{e}}) + \frac{1}{\sigma_{\mathrm{w}}^2}\mathcal{L}_{plan}(\mathrm{p}_{pred}, \hat{\mathrm{p}}_{\mathrm{w}}) + \log(\sigma_{\mathrm{w}}) \\ + \mathcal{L}_{cls}(\mathrm{c}_{pred,a}, \mathrm{c}_{\mathrm{a}}) + \frac{1}{\sigma_{\mathrm{w,a}}^2}\mathcal{L}_{mot}(\mathrm{p}_{pred,a}, \hat{\mathrm{p}}_{\mathrm{w,a}}) \\ + \log(\sigma_{\mathrm{w,a}}) + \mathcal{L}_{tpt}(\mathrm{c}_{pred}, \mathrm{c}_{\mathrm{p}}, \mathrm{c}_{\mathrm{n}}) + \mathcal{L}_{tpt}(\mathrm{c}_{\mathrm{a}}, \mathrm{c}_{\mathrm{p,a}}, \mathrm{c}_{\mathrm{n,a}}) \\ + D_{KL}(\mathrm{c}_{pred,e}||\mathrm{c}_{\mathrm{e}}) + D_{KL}(\mathrm{c}_{pred,a}||\mathrm{c}_{\mathrm{a}}),\end{aligned} \tag{7}$$

where p_{pred}, c_{pred}, $\mathrm{p}_{pred,a}$, 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. This loss encourages the base model's prediction to align with the probabilistic prediction by the trained Gaussian process, which improves prediction robustness and regularizes against noise in training data.

Adaptation in Target Domain When 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 in Eq. 6 is used together with the GP-based regularization in Eq. 7. The final loss $\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{gp}$.

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). As another example, in an online setting, the E2E driving model streams through video frames as the car drives and does not have access to the ground-truth waypoints on the fly. For unsupervised domain adaptation, as ground-truth labels are not available, we use $\mathcal{L}_{gp}$ to update the base E2E model.

4 Experiments

We conduct a thorough evaluation of RoCA in both closed-loop and open-loop planning settings. Our experiments demonstrate the domain adaptation capabilities of RoCA as a general plug-and-play framework that can be integrated with different end-to-end (E2E) autonomous driving models. Specifically, we present closed-loop evaluations on Bench2Drive, followed by cross-domain experiments that include simulation-to-real transfer from Bench2Drive to nuScenes, cross-city adaptation using nuScenes, and domain generalization under image degradations such as adverse weather, low-light conditions, and motion blur. In addition, we evaluate RoCA in an active learning setting for cross-domain adaptation, leveraging uncertainty estimates from our GP-based formulation.

4.1 Experimental Setup

Datasets. We use Bench2Drive [17] for closed-loop evaluation, which leverages the CARLA simulator and features 44 difficult interactive scenarios across diverse

Table 1: Closed-loop evaluation on the challenging 220 routes of Bench2Drive [17]. "L" means large configuration for the corresponding model.

Method	Closed-loop metric			Open-loop metric	Ability (%)↑					
	DS↑	SR ↑	Efficiency↑	Avg. L2↓	Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	Mean
VAD [20]	42.35	15.0	157.94	0.91	8.11	24.44	18.64	20.00	19.15	18.07
SSR [23]	47.71	24.4	97.41	0.86	19.10	29.38	41.67	30.00	56.75	35.38
SparseDrive-S [36]	51.01	27.8	103.1	0.83	22.64	30.38	44.56	30	53.81	36.28
GenAD [48]	44.81	15.9	–	-	-	-	-	-	-	-
DriveTransformer-L [18]	63.46	35.0	100.64	0.62	17.57	35.00	48.36	40.00	52.10	38.60
ORION [10]	77.74	54.6	151.48	0.68	25.00	71.11	78.33	30.00	69.15	54.72
RoCA (VAD)	56.90	34.3	175.42	0.74	27.39	43.91	53.76	30	47.17	40.45
RoCA (SSR)	59.81	41.0	110.61	0.69	31.96	48.23	60.94	40	63.75	48.98
RoCA (ORION)	80.38	58.2	181.06	0.57	34.06	74.91	83.76	40	72.83	61.11

weather and urban conditions.[2] We utilize its official base training set (1,000 clips) for a fair comparison with other baselines, and evaluate performance on 220 challenging routes. To assess open-loop planning, we use the nuScenes dataset [1], which consists of 28,000 samples with 22k/6k training/validation splits. nuScenes contains data collected from Boston and Singapore, which allows us to evaluate cross-domain generalization and adaptation across cities. Within the nuScenes validation set, we also consider a "targeted" subset containing 689 samples where the vehicle must make a turn, as established in [44]. To further evaluate robustness, we also use degraded versions of the validation set using controlled image corruptions such as adverse weather (like snow and fog), motion blur and low light from [47].

Evaluation protocol. For closed-loop evaluation metrics, we adopt the protocol proposed by [18], and evaluate the driving score (DS) and efficiency on Dev10. For open-loop evaluation metrics, we adopt the standardized evaluation proposed by [44] to compute two metrics, the L2 error (in meters) between the predicted and ground-truth waypoints, and the collision rate (in %) between the ego-vehicle and the surrounding vehicles. We compute open-loop metrics for comparisons on Bench2Drive and nuScenes validation.

Model details. We consider four recent and representative E2E planning models as our base architectures: ORION [10], SSR [23], SparseDrive [36], and VAD [20]. In our experiments, we adopt the *tiny* configuration of VAD (VAD-Tiny) and the *small* configuration of SparseDrive (SparseDrive-S). Note that our proposed RoCA can be used with any E2E planning model, as long as it provides a tokenized representation. When there are no explicit ego/agent tokens, we can apply RoCA over the queries or tokens that are fed into the planner. When we implement the codebook in RoCA, for ego tokens, we use 16 groups for each of the driving commands: turn left, turn right, and go straight, resulting in $N_{ego} = 48$. We use $N_{agent} = 64$ groups to capture various types of agent trajectories. The total groups in the codebook is $N_{code} = N_{ego} + N_{agent} = 112$ and we set the group size $C = 64$. Each basis has dimension of $D = 256$.

4.2 Closed-Loop Evaluations on Bench2Drive

Table 1 shows the results on the challenging 220 routes of Bench2Drive. In this case, we only use RoCA as a training regularization and no adaptation is

[2] See Table 2 in Bench2Drive paper for more details

Table 2: Sim-to-real model performance in zero-shot and fine-tuned settings from Bench2Drive [17] to nuScenes [1]. For fine-tuned results, values in parentheses denote our domain adaptation performance without using ground-truth labels.

Method	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-Tiny [20]	1.32	0.51	1.59	0.54	0.91	0.39	1.27	0.39
SSR [23]	1.08	0.31	1.47	0.44	0.75	0.15	1.19	0.37
SparseDrive-S [36]	1.17	0.34	1.51	0.49	0.65	0.14	0.85	0.31
DiMA [12]	0.94	0.26	1.29	0.38	0.61	0.19	0.94	0.30
DriveTransformer-S [18]	1.12	0.33	1.44	0.43	0.69	0.20	0.95	0.33
ORION [10]	0.98	0.44	1.56	0.51	0.72	0.29	1.12	0.37
RoCA (VAD-Tiny)	0.85	0.24	1.19	0.34	0.63 (0.73)	0.12 (0.17)	0.88 (0.95)	0.24 (0.27)
RoCA (SSR)	0.79	0.22	1.10	0.32	0.54 (0.63)	0.09 (0.13)	0.65 (0.77)	0.25 (0.29)
RoCA (SparseDrive-S)	0.75	0.22	0.98	0.30	0.55 (0.64)	0.10 (0.15)	0.64 (0.80)	0.24 (0.27)
RoCA (ORION)	0.68	0.23	0.94	0.30	0.44 (0.52)	0.08 (0.11)	0.56 (0.72)	0.20 (0.24)

Table 3: Cross-domain closed-loop evaluation results. Higher is better.

(a) Bench2Drive → NAVSIM

Method	NC↑	EP↑	PDMS↑
SparseDrive	97.1	74.2	82.1
Orion	97.9	78.7	84.4
RoCA-SparseDrive	98.0	77.5	85.8
RoCA-Orion	**98.4**	**82.6**	**87.9**

(b) Bench2Drive → DriveArena

Method	RC↑	PDMS↑
SparseDrive	13.7	68.3
Orion	17.9	71.6
RoCA-SparseDrive	28.1	84.4
RoCA-Orion	**33.7**	**91.2**

conducted during test time. We see that by using RoCA, we significantly improve the planning performance, e.g., improving driving score from 77.76 to 80.38 with ORION as the base model. This table highlights similar trends: RoCA (ORION) improves mean ability by 11.7% over the baseline ORION (i.e. 54.72 → 61.11), while RoCA (SSR) delivers a 27.8%(i.e. 35.38 → 48.98) improvement over SSR. Performance gains span critical skills such as merging, overtaking, emergency braking, yielding, and traffic-sign compliance—underscoring that GP module of RoCA predicts better posterior for ego waypoint trajectories.

4.3 Sim-to-Real Evaluation

We conduct a sim-to-real experiment by transferring models trained on Bench2Drive to the nuScenes dataset. The base E2E model, state-of-the-art baselines, and the proposed RoCA module are first trained on the source domain (Bench2Drive) for 12 epochs. In the target domain (nuScenes), we evaluate both zero-shot performance (*i.e.* no adaptation) and performance after 12 epochs of adaptation/fine-tuning. Specifically, we compare: (i) the baseline model, (ii) RoCA with source-only training regularization, and (iii) adaptation approaches with and without ground-truth labels. We also report results on the more challenging targeted split of nuScenes. Table 2 shows that RoCA consistently achieves superior sim-to-real performance compared to baselines and the LLM-based ORION model. For example, in zero-shot settings on the full validation split, RoCA improves average L2 error from 0.98 (ORION) to 0.68, and on the targeted split from 1.56 to 0.94. After fine-tuning, RoCA further reduces error to 0.44 (0.51 without ground truth)

on full validation and 0.56 (0.72 without ground truth) on the targeted split, outperforming ORION and other baselines. Moreover, even without ground-truth labels during adaptation, RoCA maintains strong performance (values in parentheses), demonstrating its robustness and effectiveness in unsupervised domain adaptation.

4.4 Closed-Loop Cross-Domain Evaluation on NAVSIM

To further evaluate performance under realistic interactive conditions, we conduct a closed-loop cross-domain evaluation by training on Bench2Drive (source) and evaluating on NAVSIM v1 [7] (target). We adopt planning-oriented metrics—No-at-fault Collisions (NC↑), Ego Progress (EP↑), and PDMS↑—which better reflect closed-loop performance compared to displacement-based open-loop metrics. For RoCA-Orion and RoCA-SparseDrive, we additionally perform unsupervised fine-tuning following Sections 3.2 and 3.3. As shown in Table 3 (a), RoCA consistently improves performance across all metrics. In particular, RoCA improves PDMS by +3.7 over SparseDrive (85.8 vs. 82.1) and by +3.5 over Orion (87.9 vs. 84.4), while also achieving gains in EP (+2.3 / +3.2) and NC (+0.9 / +0.5). These results demonstrate stronger closed-loop planning under domain shift, including robustness to differences in sensor configurations and interaction dynamics. Evaluating on NAVSIM provides a more realistic assessment beyond nuScenes, and we plan to extend this analysis to NAVSIM v2 in future work.

4.5 Closed-Loop Evaluation in DriveArena

To further evaluate performance under realistic interactive conditions, we conduct closed-loop cross-domain experiments using the DriveArena [49] simulator. Specifically, models are trained on Bench2Drive (source) and evaluated zero-shot on DriveArena (target). We report Route Completion (RC↑) and PDMS↑, which measure task success and planning quality in closed-loop environments. Table 3 (b) shows that RoCA consistently improves performance over the corresponding base planners. For example, RoCA-Orion improves route completion from 17.9 to 33.7 and PDMS from 71.6 to 91.2, demonstrating substantial gains in interactive driving performance under domain shift. Importantly, RoCA is plug-and-play for real-world simulators such as DriveArena. The Gaussian Process module is used only during training and adaptation, and the deployed planner remains unchanged, incurring no additional inference latency.

4.6 Cross-City Evaluation

Table 4 summarizes the results of transferring models across cities for evaluating cross-domain performance. When performing zero-shot inference in the target city without adaptation, RoCA performs more robustly as compared to the baseline. For instance, using VAD-Tiny as the baseline and running Boston-trained models in Singapore, the collision rate of RoCA is less than half of VAD-Tiny (0.211% vs. 0.430%). When adapting with ground truth, RoCA significantly outperforms

Table 4: Cross-city planning performance on nuScenes dataset [1].

Method	Adapt	Boston → Singapore		Singapore → Boston	
		Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓
Senna [19]	✗	1.03	0.26	0.96	0.24
w/ finetuning using GT	✓	0.98	0.23	0.69	0.19
DiMA [12]	✗	1.15	0.23	0.92	0.22
w/ finetuning using GT	✓	1.01	0.21	0.72	0.17
VAD-Tiny [20]	✗	1.25	0.43	1.16	0.23
w/ finetuning using GT	✓	1.19	0.26	1.11	0.19
w/ RoCA training regularization	✗	1.17	0.21	1.01	0.19
w/ RoCA unsupervised adaptation	✓	1.02	0.19	0.94	0.17
w/ RoCA adaptation using GT	✓	0.94	0.17	0.89	0.16
SparseDrive-S [36]	✗	0.91	0.18	1.02	0.24
w/ finetuning using GT	✓	0.55	0.12	0.67	0.12
w/ RoCA training regularization	✗	0.79	0.15	0.88	0.15
w/ RoCA unsupervised adaptation	✓	0.71	0.13	0.68	0.11
w/ RoCA adaptation using GT	✓	0.49	0.09	0.51	0.10

Table 5: 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

direct finetuning across all metrics. For example, on Boston → Singapore transfer using SparseDrive, RoCA reduces L2 error from 0.55 m to 0.49 m and collision rate from 0.12% to 0.09%. This setting provides the fairest comparison, as both approaches utilize ground-truth supervision. Even when unsupervised, RoCA still outperforms direct finetuning with ground truth in most cases, highlighting its strong domain adaptation capability and the benefit of our probabilistic modeling.

4.7 Active Learning

In the target domain, active learning reduces annotation and adaptation costs if the the most informative samples are identified and used for training. To achieve this, we propose using the GP-based predictive variance as a uncertainty criterion, selecting samples with the highest estimated variance. We compare our variance-based selection with the random sampling baseline, evaluated at 5%, 10%, and 15% sampling rates of the full target training data. Table 5 reports cross-city transfer results between Singapore and Boston after fine-tuning using ground-truth supervision, with SparseDrive-S as the base model. Across all sampling rates, uncertainty-based selection with RoCA consistently improves planning performance over random sampling. Furthermore, RoCA consistently

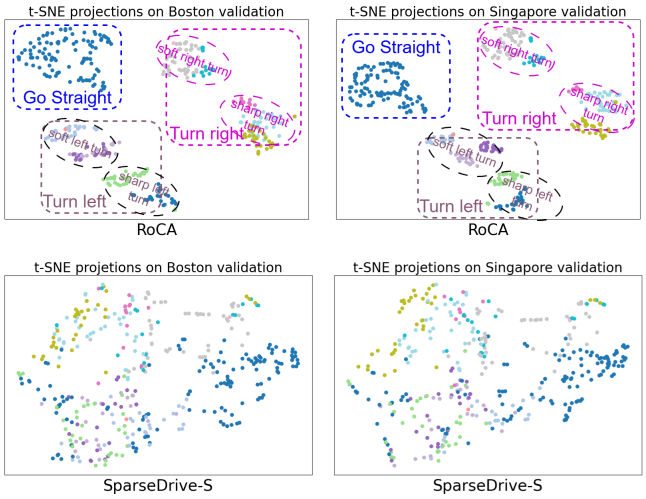

Fig. 2: tSNE projection of ego/agent tokens with (top) and without (bottom) RoCA. By using our proposed approach, the model has better separability of different trajectory modes (indicated by different colors). In contrast, the baseline SparseDrive shows poor separability, indicating a sensitivity to any perturbations. The analysis is performed on the full nuScenes val set, the Boston and Singapore subsets(left, middle, right). Models are trained on a source city and evaluated using t-SNE projections on a target city. For example, in the left column, the model is trained on Singapore and plotted on Boston validation.

outperforms the baseline under both sampling strategies, underscoring its better adaptation capability.

4.8 Learned Tokens in GP

Our GP-based formulation in RoCA yields a more structured and semantically meaningful token space for ego and agent trajectories, leading to robust generalization across domains. As illustrated in Figure 2, RoCA produces well-separated clusters that correspond to distinct trajectory modes, whereas the baseline SparseDrive-S exhibits significant overlap and entanglement, making it sensitive to perturbations or noises to tokens to this token (*e.g.* due to different camera characteristics, lighting, etc.). The improved separability is further evident in Figure 2, where commands like *Go Straight*, *Turn Left*, and *Turn Right* form coherent groups, with sub-clusters capturing fine-grained variations (e.g., soft vs. sharp turns). This organization arises from the GP's kernel-based similarity structure and its uncertainty-aware regularization, enabling RoCA to maintain stable behavior and infer consistent trajectories even under domain shift.

4.9 Adaption to Image Degradations

We further assess the generalization and adaptation performance when the image quality is compromised, *e.g.* due to low light, motion blur, snow, and fog. Table 6

Table 6: Planning performance under image degradations including low light, motion blur, and under adverse weather conditions.

Model	Adapt	lowlight		motion blur		Snow		Fog	
		Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓	Avg. L2 (m) ↓	Avg. Col.(%)↓
Full Val									
SparseDrive-S	✗	0.577	0.345	0.729	0.369	0.857	0.192	0.809	0.349
w/ supervision using GT	✓	0.547	0.228	0.573	0.118	0.610	0.111	0.704	0.272
w/ RoCA training regularization	✗	0.564	0.129	0.671	0.208	0.729	0.173	0.750	0.253
w/ RoCA unsupervised adaptation	✓	0.531	0.098	0.589	0.146	0.628	0.121	0.682	0.173
w/ RoCA supervised adaptation using GT	✓	0.526	0.090	0.541	0.104	0.581	0.096	0.619	0.151
Targeted Val									
SparseDrive-S	✗	0.712	0.325	0.832	0.389	0.907	0.287	0.939	0.399
w/ supervision using GT	✓	0.707	0.291	0.721	0.282	0.729	0.188	0.772	0.256
w/ RoCA training regularization	✗	0.703	0.228	0.766	0.248	0.849	0.208	0.853	0.220
w/ RoCA unsupervised adaptation	✓	0.626	0.188	0.733	0.203	0.756	0.135	0.721	0.166
w/ RoCA supervised adaptation using GT	✓	0.591	0.169	0.696	0.192	0.660	0.119	0.641	0.141

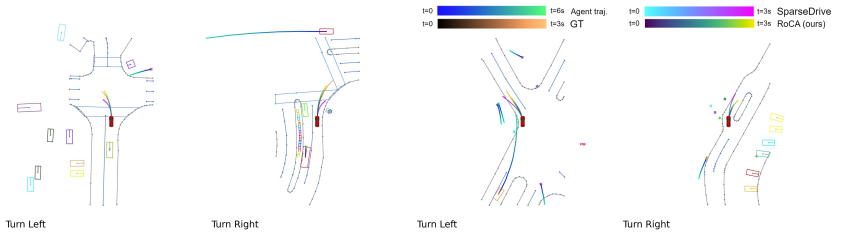

Fig. 3: Visualization of sample planning results. Left (Right) two scenarios are in Boston (Singapore); note that t is right (left) driving in Boston (Singapore). The red car is the ego vehicle. The color gradient indicates the temporal horizon of the trajectory.

shows that, without adaptation, the baseline SparseDrive-S has significantly worse performance under such adverse conditions, whereas the model trained with our RoCA regularization performs more robustly. When adaptation is performed, our RoCA unsupervised adaptation achieves significant improvement, and is on par with or better than the baseline adaptation which trains the model using ground-truth data with the adverse-conditioned images. When we further utilize the ground truth in adaptation, RoCA achieves significantly better planning performance. These results confirm that RoCA strengthens generalization and adaptation under diverse degradations, improving the reliability of E2E planning.

4.10 Qualitative Results

Figure 3 shows qualitative planning results on turning scenarios. Our RoCA approach generates trajectories that closely align with the ground-truth trajectories. On the other hand, the baseline SparseDrive model produces trajectories that will lead the vehicle into non-drivable areas.

5 Conclusions

We introduced RoCA, a robust cross-domain E2E autonomous driving framework built on a Gaussian Process formulation that jointly models ego and agent trajectories. This probabilistic structure enables uncertainty-aware planning, improves source-domain training through GP-based regularization, and significantly enhances generalization to unseen environments. A key strength of RoCA is its flexible adaptation capability, supporting standard finetuning, uncertainty-guided active learning, and online adaptation for practical deployment.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
2. Casas, S., Sadat, A., Urtasun, R.: Mp3: A unified model to map, perceive, predict and plan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14403–14412 (2021)
3. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In: Kaelbling, L.P., Kragic, D., Sugiura, K. (eds.) Proceedings of the Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 100, pp. 86–99. PMLR (30 Oct–01 Nov 2020), https://proceedings.mlr.press/v100/chai20a.html
4. Chen, D., Krähenbühl, P.: Learning from all vehicles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17222–17231 (2022)
5. Chitta, K., Prakash, A., Geiger, A.: Neat: Neural attention fields for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15793–15803 (2021)
6. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9329–9338 (2019)
7. Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., et al.: Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. Advances in Neural Information Processing Systems **37**, 28706–28719 (2024)
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
9. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C.R., Zhou, Y., 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. pp. 9710–9719 (2021)
10. Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24823–24834 (2025)
11. Ge, P., Sun, Y.: Gaussian process-based transfer kernel learning for unsupervised domain adaptation. Mathematics **11**(22), 4695 (2023)
12. Hegde, D., Yasarla, R., Cai, H., Han, S., Bhattacharyya, A., Mahajan, S., Liu, L., Garrepalli, R., Patel, V.M., Porikli, F.: Distilling multi-modal large language models for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 27575–27585 (June 2025)
13. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: European Conference on Computer Vision. pp. 533–549. Springer (2022)
14. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17853–17862 (2023)

15. Huang, F., Fan, Z., Li, X., Zhang, W., Li, P., Geng, Y., Zhu, K.: Tailored meta-learning for dual trajectory transformer: advancing generalized trajectory prediction. Complex & Intelligent Systems **11**(3), 174 (2025)
16. Hwang, J.J., Xu, R., Lin, H., Hung, W.C., Ji, J., Choi, K., Huang, D., He, T., Covington, P., Sapp, B., et al.: Emma: End-to-end multimodal model for autonomous driving. arXiv preprint arXiv:2410.23262 (2024)
17. Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Advances in Neural Information Processing Systems **37**, 819–844 (2024)
18. Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In: International Conference on Learning Representations (ICLR) (2025)
19. Jiang, B., Chen, S., Liao, B., Zhang, X., Yin, W., Zhang, Q., Huang, C., Liu, W., Wang, X.: Senna: Bridging large vision-language models and end-to-end autonomous driving. arXiv preprint arXiv:2410.22313 (2024)
20. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023)
21. Kim, M., Sahu, P., Gholami, B., Pavlovic, V.: Unsupervised visual domain adaptation: A deep max-margin gaussian process approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4380–4390 (2019)
22. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
23. Li, P., Cui, D.: Navigation-guided sparse scene representation for end-to-end autonomous driving. In: International Conference on Learning Representations. vol. 2025, pp. 29118–29134 (2025)
24. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
25. Liang, C., Chen, W., Zhao, X., Wang, J., Cao, L., Zhang, J.: Distribution optimization under gaussian hypothesis for domain adaptive semantic segmentation. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 9280–9290. IEEE (2025)
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36** (2024)
27. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7577–7586 (2021)
28. Loh, J.Q., Luo, X., Ding, F., Tew, H.H., Loo, J.Y., Ding, Z.Y., Susilawati, S., Tan, C.P.: Cross-domain transfer learning using attention latent features for multi-agent trajectory prediction. In: 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 3905–3911. IEEE (2024)
29. Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting multiple agent trajectories. International Conference on Learning Representations (2022)

30. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020)
31. Qian, T., Chen, Y., Cong, G., Xu, Y., Wang, F.: Adaptraj: A multi-source domain generalization framework for multi-agent trajectory prediction. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE). pp. 5048–5060. IEEE (2024)
32. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
33. Seeger, M.W.: Gaussian processes for machine learning. International journal of neural systems **14 2**, 69–106 (2004), https://api.semanticscholar.org/CorpusID:283284267
34. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Beißwenger, J., Luo, P., Geiger, A., Li, H.: Drivelm: Driving with graph visual question answering. In: European conference on computer vision. pp. 256–274. Springer (2024)
35. Sun, Q., Zhang, S., Ma, D., Shi, J., Li, D., Luo, S., Wang, Y., Xu, N., Cao, G., Zhao, H.: Large trajectory models are scalable motion predictors and planners. arXiv preprint arXiv:2310.19620 (2023)
36. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8795–8801. IEEE (2025)
37. Tian, T., Li, B., Weng, X., Chen, Y., Schmerling, E., Wang, Y., Ivanovic, B., Pavone, M.: Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 3656–3673. PMLR (06–09 Nov 2025), https://proceedings.mlr.press/v270/tian25b.html
38. Tian, X., Gu, J., Li, B., Liu, Y., Hu, C., Wang, Y., Zhan, K., Jia, P., Lang, X., Zhao, H.: Drivevlm: The convergence of autonomous driving and large vision-language models. arXiv preprint arXiv:2402.12289 (2024)
39. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Álvarez, J.M.: Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
40. Wang, W., Xie, J., Hu, C., Zou, H., Fan, J., Tong, W., Wen, Y., Wu, S., Deng, H., Li, Z., et al.: Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. arXiv preprint arXiv:2312.09245 (2023)
41. Wang, Z., Guo, J., Zhang, H., Wan, R., Zhang, J., Pu, J.: Bridging the gap: Improving domain generalization in trajectory prediction. IEEE Transactions on Intelligent Vehicles **9**(1), 1780–1791 (2023)
42. Wang, Z., Huang, Z., Gao, Y., Wang, N., Liu, S.: Mv2dfusion: Leveraging modality-specific object semantics for multi-modal 3d detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
43. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
44. Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: Para-drive: Parallelized architecture for real-time autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15449–15458 (2024)

45. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA (2006)
46. Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: 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)
47. Xie, S., Kong, L., Zhang, W., Ren, J., Pan, L., Chen, K., Liu, Z.: Benchmarking and improving bird's eye view perception robustness in autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **47**(5), 3878–3894 (2025)
48. Yang, J., Gao, S., Qiu, Y., Chen, L., Li, T., Dai, B., Chitta, K., Wu, P., Zeng, J., Luo, P., Zhang, J., Geiger, A., Qiao, Y., Li, H.: Generalized predictive model for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
49. Yang, X., Wen, L., Wei, T., Ma, Y., Mei, J., Li, X., Lei, W., Fu, D., Cai, P., Dou, M., et al.: Drivearena: A closed-loop generative simulation platform for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26933–26943 (2025)
50. Yasarla, R., Patel, V.M.: Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8405–8414 (2019)
51. Yasarla, R., Perazzi, F., Patel, V.M.: Deblurring face images using uncertainty guided multi-stream semantic networks. IEEE Transactions on Image Processing **29**, 6251–6263 (2020)
52. Yasarla, R., Priebe, C.E., Patel, V.M.: Art-ss: an adaptive rejection technique for semi-supervised restoration for adverse weather-affected images. In: European Conference on Computer Vision. pp. 699–718. Springer (2022)
53. Yasarla, R., Sindagi, V.A., Patel, V.M.: Syn2real transfer learning for image deraining using gaussian processes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2726–2736 (2020)
54. Yasarla, R., Sindagi, V.A., Patel, V.M.: Semi-supervised image deraining using gaussian processes. IEEE Transactions on Image Processing **30**, 6570–6582 (2021)
55. Yasarla, R., Sindagi, V.A., Patel, V.M.: Unsupervised restoration of weather-affected images using deep gaussian process-based cyclegan. In: 2022 26th International Conference on Pattern Recognition (ICPR). pp. 1967–1974. IEEE (2022)
56. Yasarla, R., Valanarasu, J.M.J., Sindagi, V., Patel, V.M.: Self-supervised denoising transformer with gaussian process. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1474–1484 (2024)
57. Ye, L., Zhou, Z., Wang, J.: Improving the generalizability of trajectory prediction models with frenet-based domain normalization. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 11562–11568. IEEE (2023)
58. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)