Pioneering SE(2)-Equivariant Trajectory Planning for Automated Driving

Steffen Hagedorn[1], Marcel Milich[2], and Alexandru P. Condurache[1]

Abstract— **Planning the trajectory of the controlled ego vehicle is a key challenge in automated driving. As for human drivers, predicting the motions of surrounding vehicles is important to plan the own actions. Recent motion prediction methods utilize equivariant neural networks to exploit geometric symmetries in the scene. However, no existing method combines motion prediction and trajectory planning in a joint step while guaranteeing equivariance under roto-translations of the input space. We address this gap by proposing a lightweight equivariant planning model that generates multi-modal joint predictions for all vehicles and selects one mode as the ego plan. The equivariant network design improves sample efficiency, guarantees output stability, and reduces model parameters. We further propose equivariant route attraction to guide the ego vehicle along a high-level route provided by an off-the-shelf GPS navigation system. This module creates a momentum from embedded vehicle positions toward the route in latent space while keeping the equivariance property. Route attraction enables goal-oriented behavior without forcing the vehicle to stick to the exact route. We conduct experiments on the challenging nuScenes dataset to investigate the capability of our planner. The results show that the planned trajectory is stable under roto-translations of the input scene which demonstrates the equivariance of our model. Despite using only a small split of the dataset for training, our method improves L2 distance at $3\,\mathrm{s}$ by $20.6\,\%$ and surpasses the state of the art.**

I. INTRODUCTION

In automated driving, trajectory planning is the task of finding a safe and efficient path and speed profile of a controlled ego vehicle (EV) toward a goal position [1]. In addition to past positions, map, and route information, many planning methods rely on motion prediction of surrounding vehicles (SVs) to model interactions [1]–[4]. Combining prediction and planning by handling all vehicles jointly is a promising approach to reduce computation and overcome purely reactive behavior [2], [5], [6]. To increase sample efficiency and robustness, the predicted trajectories of all vehicles must be independent of the viewpoint from which the scene is observed (cf. Fig. 1) [7]. Equivariant models fulfill this requirement and are therefore utilized in many tasks that solve problems in observable physical systems [8]–[10]. Equivariance means that transformations of the input transform the output in an equivalent way. Designing equivariant neural networks (NNs) is beneficial in multiple ways. Beside guaranteeing output stability they have an increased sample efficiency and can reduce model parameters [11].

[1]Robert Bosch GmbH, 71229 Leonberg, Germany and Institute for Signal Processing, Universität zu Lübeck, 23562 Lübeck, Germany. `steffen.hagedorn@de.bosch.com`, [2]Bosch Center for Artificial Intelligence, 71272 Renningen, Germany and Institute for Parallel Distributed Systems, Universität Stuttgart, 70569 Stuttgart, Germany
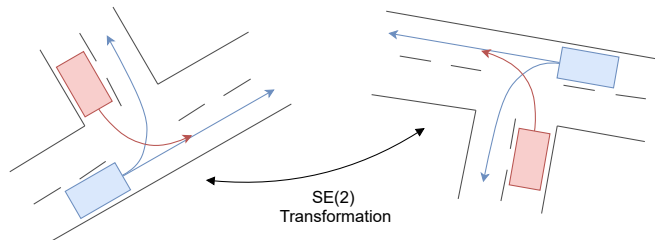
Fig. 1: Exemplary traffic scene that demonstrates the intuition behind SE(2)-equivariant trajectory prediction and planning: Roto-translations of the input scene should result in an equivalent transformation of the trajectory output.

When given a fixed-size dataset, inducing prior knowledge by means of equivariance can increase the performance [12].

While methods for joint prediction and planning have been presented and the advantages of equivariance have been used in stand-alone motion prediction, no existing planning model combines both techniques.

Instead, many methods transform the whole scene into a coordinate system centered around one vehicle, typically the ego vehicle [13]–[15]. However, this approach has proven sample-inefficient and vulnerable to domain shifts as the scene representation is viewpoint-dependent [7], [16]. Other works alleviate this problem by taking the perspective of each vehicle [17]–[19]. Such methods are more robust but computationally expensive [20] which also is a known downside of universally applicable equivariant approaches that are based on irreducible representations of the transformation group [21], [22]. In contrast, EqMotion reduces the computation by explicitly designed equivariant operations that do not rely on irreducible representations [9]. EqDrive applies EqMotion for vehicle motion prediction in traffic scenes [23]. Their results demonstrate that equivariant neural networks can improve the performance in automated driving tasks.

Since the benefits of joining prediction and planning as well as the advantages of equivariant models are shown in recent methods, we want to pioneer the field of combining both aspects. Therefore we propose PEP (Pioneering Equivariant Planning): A lightweight equivariant planning method that integrates prediction and planning in a joint approach. Similar to EqMotion [9], PEP is a graph neural network [24] that consists of an equivariant feature learning branch that processes vehicle positions and an invariant branch that processes invariant features such as the distance between two positions. We extend the architecture by adding another equivariant branch that updates the EV position by providing route information. This allows the joint processing of EV and

SVs while conditioning the EV prediction on a goal. We further add a mode selection layer that selects the EV plan from K predicted modes. The network is trained with a loss that jointly optimizes planning and prediction, and promotes diverse multi-modal predictions.

We evaluate PEP on the prediction split of nuScenes [25]. Alongside the open-loop evaluation, we present a comprehensive ablation study and equivariance analysis.

In summary, our contributions are:
- We present the first equivariant planner integrated with multi-modal joint prediction.
- We propose an equivariant route attraction mechanism that allows following a high-level route.
- We report state-of-the-art performance on the nuScenes dataset in open-loop planning.

II. RELATED WORK

A. Joint Prediction and Planning

Early planning models for automated driving plan the EV's trajectory directly from perception inputs without explicitly considering the interplay with SVs [6]. Alternatively, many solutions sequentially employ separate subsystems for prediction and planning [3], [26]–[28]. While increasing explainability, such methods still handle EV and SVs separately and lead to reactive behavior [6]. By modeling the future of all vehicles simultaneously, joint prediction and planning goes beyond reactive behavior and can reduce computation [2], [5]. Joint prediction and planning approaches can be categorized into iterative methods and regression.

The iterative probabilistic method PRECOG predicts the state of all vehicles one step into the future and uses the outcome as input for the next iteration [2]. Goal information is provided for the EV and leads to more precise predictions for all vehicles. The EV's plan is then inferred by optimizing the expectation of the predicted distribution. GameFormer is another iterative approach based on game theory [5]. Interactions are modeled as a level-k game in which the individual predictions are updated based on the predicted behavior of all vehicles from the previous level. The encoder-decoder transformer architecture predicts multiple modes for SVs while restricting EV prediction to a single mode that serves as the plan.

In contrast, regressive methods learn a joint feature for the whole prediction horizon from which complete trajectories are regressed. SafePathNet employs a transformer for multi-modal joint prediction of EV and SVs [29]. Every predicted EV mode is then checked for collisions with the most probable mode of each SV. The EV mode with the lowest predicted collision rate is selected as the plan. Similarly, DIPP starts with a multi-modal joint prediction and selects the mode with the highest probability for each vehicle [30]. To infer the EV plan, a differentiable nonlinear optimizer refines the EV prediction under consideration of the SV predictions and additional hand-crafted constraints.

We also base our planner on multi-modal joint prediction for all vehicles but further design the whole network to be equivariant under 2D roto-translations of the input.

B. Equivariant Motion Prediction

Equivariant convolutional neural networks add rotation equivariance to the inherent translation equivariance of the convolution operation [31]. Rotation equivariance in the 2D image domain is achieved by oriented convolutional filters [32], log-polar coordinates [33], circular harmonics [34] or steerable filters [12], [35]. With the advent of Graph Neural Networks (GNNs) [24] which work on sparse data representations, equivariant adaptions of this architecture emerged. To extract roto-translation-equivariant features from point clouds or graphs, some approaches utilize irreducible representations of the transformation group such as spherical harmonics [36], [37]. Others base their method on specifically designed and computationally less expensive layers [38]–[40]. For example, [39] embed the representation of neurons into a 3D space where they leverage basic principles of matrix algebra to induce equivariance. Equivariant GNNs are a common choice for solving tasks in physical systems since these often possess rotational and translational symmetries [41], [42]. Since motion prediction models a physical system, equivariant NNs are well-suited for this task. [43] first apply learning-based equivariant motion prediction to predict fluid flow. SE(3)-equivariant transformers mark another milestone by regressing pedestrian trajectories [37]. The first equivariant motion prediction model for autonomous driving utilizes polar coordinate-indexed filters to design an equivariant continuous convolution operator [10]. Recently, the motion prediction network EqMotion presents strong performance on various tasks [9]. Similar to [39], its specifically designed layers exploit geometrical symmetries. In addition to the equivariant feature learning they present an invariant interaction reasoning module and an invariant pattern feature. This design allows to integrate prior knowledge efficiently. Features like absolute distances which are inherently SE(3)-invariant can be processed in the invariant layers while absolute positions are handled by the SE(3)-equivariant layers. EqDrive finally applies EqMotion for vehicle trajectory prediction [9]. In this work, we propose an equivariant model based on EqMotion that extends the motion prediction to a trajectory planner for the EV and integrates prior knowledge of its intended route.

III. METHOD

In this section, we introduce our equivariant trajectory planner PEP, which is an expansion of EqMotion. For a more detailed overview of EqMotion, we refer the reader to [9]. Fig. 2 provides an overview of our model.

A. Problem Formulation

PEP is an equivariant trajectory planner based on multi-modal joint prediction of all vehicles and trained by imitation learning. Given the past trajectories $\mathbf{X}_i = [x_i^1, x_i^2, \ldots, x_i^{T_p}] \in \mathbb{R}^{T_p \times 2}$ of $i = 1, \ldots, M$ vehicles, including the EV, and EV route information $\mathbf{L} \in \mathbb{R}^{C \times 2}$, the planning task is to forecast $\hat{\mathbf{Y}}_{EV} = [\hat{y}_{EV}^1, \hat{y}_{EV}^2, \ldots, \hat{y}_{EV}^{T_f}] \in \mathbb{R}^{T_f \times 2}$ as close to the real future trajectory $\mathbf{Y}_{EV}$ as possible. We further denote the set of all past trajectories as $\mathbb{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_M]$.
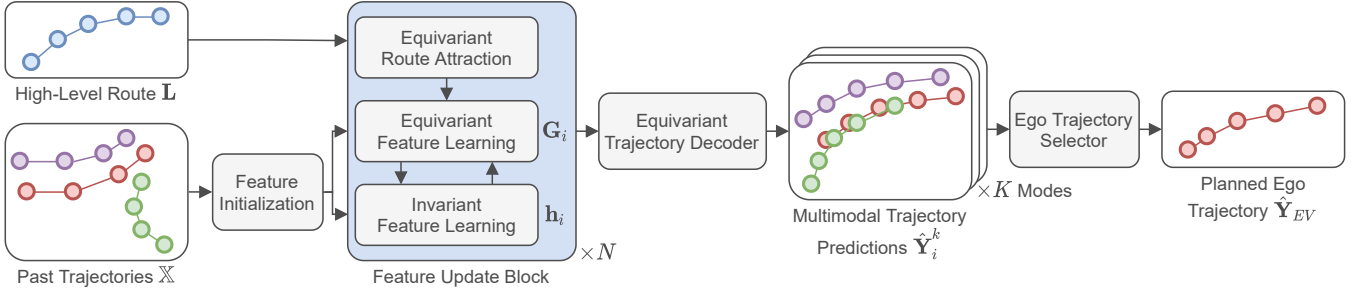
Fig. 2: PEP model overview. After feature initialization, the features are updated N times in three parallel but interacting branches. A multi-modal decoder then predicts multiple future scenarios for all vehicles jointly. Alongside the trajectories, a probability for each scenario is estimated. The EV trajectory of the most probable mode is selected as the plan.

Especially, we require the planning function $\mathcal{F}_{\text{plan}}(\mathbb{X}, L) = \hat{\mathbf{Y}}_{EV}$ to be equivariant under transformations $\mathcal{T}_g$ in the Euclidean group SE(2), which comprises rotations $\mathbf{R} \in SO(2)$ and translations $\mathbf{t} \in \mathbb{R}^2$. All feature updates f in $\mathcal{F}_{\text{plan}}$ must satisfy the equivariance condition $f(x\mathcal{T}_g) = f(x)\mathcal{T}_g$ where the roto-translation right group action $\mathcal{T}_g$ acts on 2D inputs x via matrix-multiplication and addition $x\mathcal{T}_g = x\mathbf{R} + \mathbf{t}$.

B. Feature Initialization

The key idea of handling a set of positions translation-equivariantly is to shift the viewpoint into the center, i.e. the mean coordinate $\bar{\mathbb{X}}$. To return to the initial coordinate system after a transformation, $\bar{\mathbb{X}}$ is re-added. Like EqMotion [9], we initialize the equivariant feature of vehicle i as

$$\mathbf{G}_i^{(0)} = \phi_{\text{init}_g}(\mathbf{X}_i - \bar{\mathbb{X}}) + \bar{\mathbb{X}} \in \mathbb{R}^{C \times 2} \quad (1)$$

where function ϕ_{init_g} is realized by a fully connected layer (FCL) [9]. In the following, all ϕ describe FCLs. Since an FCL is a linear transformation and can be expressed as a matrix multiplication, rotation-equivariance follows from the multiplicative associative law. We initialize the invariant feature of vehicle i as a function of velocity $\Delta\mathbf{X}_i$ and heading angle, which are both inferred from positions $\mathbf{X}_i$ as in EqMotion [9]. The $[\cdot; \cdot]$ operator denotes concatenation.

$$\mathbf{h}_i^{(0)} = \phi_{\text{init}_h}([||\Delta\mathbf{X}_i||_2; \text{angle}(\Delta\mathbf{X}_i^\tau, \Delta\mathbf{X}_i^{\tau-1})]) \in \mathbb{R}^D \quad (2)$$

EqMotion further adds an invariant relationship learning, which computes $\mathbf{c}_{ij} \in [0,1]^Q$ between agents i and j from the initial equivariant and invariant feature [9]. $\mathbf{c}_{ij}$ describes the relationship of i and j in Q categories. For instance, the network could learn to extract distance, velocity differences or heading differences of i and j.

C. Feature Update

1) Equivariant Route Attraction: Many automated driving systems comprise a tactical planner or navigation system to provide a coarse intended route at lane level. We introduce a novel module called 'equivariant route attraction' to incorporate the intended EV route into the joint prediction. The intuition is to move the equivariant feature of the EV toward the high-level route $\mathbf{L}$ in latent space before considering interactions with other vehicles. This order of feature updates

prioritizes social interactions over route following, which is important for collision avoidance. Since the goal is only known for the EV ($i = 0$), we update only this feature:

$$f_{\text{ra}}: \mathbf{G}_0^{(l)} \leftarrow \mathbf{G}_0^{(l)} + \phi_{\text{ra}}^{(l)}(\mathbf{L} - \mathbf{G}_0^{(l)}) \in \mathbb{R}^{C \times 2}. \quad (3)$$

The FCL ϕ_{ra} takes vector $\mathbf{L} - \mathbf{G}_0^{(l)}$ as input, which points from the equivariant EV feature embedding toward the route. Superscript (l) denotes the l-th of N feature update blocks. We show that the route attraction module f_{ra} fulfills the equivariance condition stated in the problem formulation:

$$\begin{aligned}
f_{\text{ra}}(x\mathbf{R} + \mathbf{t}) &= \mathbf{G}_0^{(l)}\mathbf{R} + \mathbf{t} + \phi_{\text{ra}}(\mathbf{L}\mathbf{R} + \mathbf{t} - (\mathbf{G}_0^{(l)}\mathbf{R} + \mathbf{t})) \\
&= \mathbf{G}_0^{(l)}\mathbf{R} + \mathbf{t} + \phi_{\text{ra}}((\mathbf{L} - \mathbf{G}_0^{(l)})\mathbf{R}) \\
&= \mathbf{G}_0^{(l)}\mathbf{R} + \mathbf{t} + \phi_{\text{ra}}(\mathbf{L} - \mathbf{G}_0^{(l)})\mathbf{R} \\
&= (\mathbf{G}_0^{(l)} + \phi_{\text{ra}}(\mathbf{L} - \mathbf{G}_0^{(l)}))\mathbf{R} + \mathbf{t} \\
&= f_{\text{ra}}(x)\mathbf{R} + \mathbf{t} \quad \square
\end{aligned} \quad (4)$$

2) Equivariant Feature Learning: This feature update step comprises inner aggregation and neighbor aggregation. Inner aggregation updates the equivariant feature of vehicle i using a weight computed from its invariant feature. $\bar{\mathbb{G}}^{(l)}$ is the mean position of equivariant features $\mathbf{G}_i^{(l)}$ [9]:

$$\mathbf{G}_i^{(l)} \leftarrow \phi_{\text{att}}^{(l)}(\mathbf{h}_i^{(l)}) \cdot (\mathbf{G}_i^{(l)} - \bar{\mathbb{G}}^{(l)}) + \bar{\mathbb{G}}^{(l)} \in \mathbb{R}^{C \times 2} \quad (5)$$

Neighbor aggregation first defines an edge weight for each neighbor based on relationship feature $\mathbf{c}_{ij}$, equivariant, and invariant features. The i-th equivariant feature is then updated by a weighted sum over all its neighbors $\mathcal{N}_i$ [9].

$$\mathbf{e}_{ij}^{(l)} = \sum_{k=1}^{K} \mathbf{c}_{ij,k}\phi_{e,k}^{(l)}([\mathbf{h}_i^{(l)}; \mathbf{h}_j^{(l)}; ||\mathbf{G}_i^{(l)} - \mathbf{G}_j^{(l)}||_2]) \in \mathbb{R}^C \quad (6)$$

$$\mathbf{G}_i^{(l)} \leftarrow \mathbf{G}_i^{(l)} + \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij}^{(l)} \cdot (\mathbf{G}_i^{(l)} - \mathbf{G}_j^{(l)}) \in \mathbb{R}^{C \times 2} \quad (7)$$

Finally, we apply the equivariant non-linear function proposed in [9] to infer $\mathbf{G}_i^{(l+1)}$.

3) Invariant Feature Learning: The last step of the feature update in EqMotion [9] is invariant feature learning:

$$\mathbf{p}_i^{(l)} = \sum_{j \in \mathcal{N}_i} \phi_m^{(l)}([\mathbf{h}_i^{(l)}; \mathbf{h}_j^{(l)}; ||\mathbf{G}_i^{(l)} - \mathbf{G}_j^{(l)}||_2]) \in \mathbb{R}^D \quad (8)$$

$$\mathbf{h}_i^{(l+1)} = \phi_{\mathbf{h}}^{(l)}([\mathbf{h}_i^{(l)}; \mathbf{p}_i^{(l)}]) \in \mathbb{R}^D \quad (9)$$

D. Trajectory Decoding

To achieve multi-modal predictions, we introduce K parallel FCL trajectory decoders. Each decoder predicts all agents simulateneously based on their equivariant features:

$$\hat{\mathbf{Y}}_i^k = \phi_{\text{dec}}^k(\mathbf{G}_i^N - \bar{\mathbb{G}}^N) + \bar{\mathbb{G}}^N \in \mathbb{R}^{(T_f+1)\times 2} \qquad (10)$$

Note that we predict an additional output beyond prediction horizon T_f. It serves as a probability indicator for the trajectory selector, which outputs the final EV plan.

E. Trajectory Selection

We define mode probability as the mean of the spatial coordinate dimension C of the additionally predicted point.

$$\mathbf{P}_i^k = \text{mean}_C(\hat{\mathbf{Y}}_i^{k,T_f+1}) \in \mathbb{R}^K \qquad (11)$$

Selecting the most probable mode yields the EV plan:

$$\hat{\mathbf{Y}}_{EV} = \hat{\mathbf{Y}}_0^{k^*} \quad \text{where} \quad k^* = \underset{k=1,\ldots,K}{\text{argmax}} \, \mathbf{P}_0^k \qquad (12)$$

To promote mode diversity we apply a winner-takes-all (WTA) loss as described below.

F. Training Objective

In accordance with the problem statement, we focus on the planning performance in the loss function. Additionally, prediction performance for SVs is optimized in order to benefit from realistic interaction modeling:

$$\mathcal{L} = \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{wta}} + \alpha \cdot \mathcal{L}_{\text{pred}}. \qquad (13)$$

Here, $\mathcal{L}_{\text{plan}}$ is the average L2 distance between the planned EV trajectory and ground truth. $\mathcal{L}_{\text{wta}}$ considers mode selection by assigning a loss of 0 if the closest mode to the ground truth is selected correctly and else 1. $\mathcal{L}_{\text{pred}}$ is the minimal average L2 error for SVs, weighted with $\alpha = 0.1$.

IV. RESULTS & DISCUSSION

A. Implementation

All results are gathered with the same architecture using $N = 4$ feature update blocks with $Q = 4$ relationship categories, a coordinate dimension of $C = D = 64$, and $K = 6$ trajectory decoders. Past and future trajectories are encoded as $T_p = 4$ and $T_f = 6$ positions, which corresponds to $t_p = 1.5\,\text{s}$ and $t_f = 3\,\text{s}$ in the selected dataset, respectively.

B. Dataset

Since PEP performs joint prediction and planning, we use only multi-vehicle scenes in the official nuScenes prediction split, i.e. 471 training and 136 test scenes [25]. These are only 607 of 1000 total scenes. Route attraction uses the high-level route the driver was supposed to follow during data acquisition, which is provided in the CAN-Bus expansion.

C. Training Setup

PEP is implemented in PyTorch and has $1.3\,\text{M}$ trainable parameters when configured as described in A. It is trained over 400 epochs with batch size 512. We used the Adam optimizer [44] with an initial learning rate of 5×10^{-4} that decreases with a factor of 0.8 every other epoch. On a single GTX 1080Ti training to convergence takes about $1.25\,\text{h}$.

TABLE I: Planning results on nuScenes

Model	Per	GC	Vel	Acc	Traj	L2 (m) 3 s	L2 (m) Avg.	CR (%) 3 s	CR (%) Avg.
NMP [45]	✓	-	-	-	-	2.31	-	1.92	-
SA-NMP [45]	✓	-	-	-	-	2.05	-	1.59	-
FF [46]	✓	-	-	-	-	2.54	1.43	1.07	0.43
EO [47]	✓	-	-	-	-	2.78	1.60	0.88	0.33
ST-P3 [48]	✓	✓	-	-	-	2.90	2.11	1.27	0.71
UniAD [4]	✓	✓	-	-	-	1.65	1.03	0.71	0.31
DeepEM [49]	✓	✓	-	-	-	0.73	0.48	0.36	0.19
FusionAD [50]	✓	✓	✓	✓	✓	-	0.81	0.27	**0.12**
VAD-Tiny [51]	✓	✓	✓	✓	✓	0.65	0.41	0.27	0.16
VAD-Base [51]	✓	✓	✓	✓	✓	0.60	0.37	**0.24**	0.14
BEV-Planner++ [52]	✓	✓	✓	✓	✓	0.57	0.35	0.73	0.34
AD-MLP-I [15]	-	-	-	-	✓	1.48	0.97	0.83	0.49
AD-MLP-II [15]	-	-	✓	✓	✓	0.49	0.35	0.28	0.23
AD-MLP-IV [15]	-	✓	✓	✓	✓	0.41	0.29	**0.24**	0.19
PEP (Ours)	-	✓	-	-	✓	**0.32**	**0.28**	0.43	0.37

D. Planning

Planning performance is evaluated in open loop. Table I provides a broad comparison with other methods. Except for our model, results are taken from [15], [49], [50], [52]. To facilitate an overview, the methods are categorized based on model design criteria. 'Per' indicates that a method uses additional information from perception, 'GC' stands for goal conditioning of the EV, and 'Vel', 'Acc', and 'Traj' encode whether ground truth velocity, acceleration, and trajectory are provided, respectively. L2 distance between the planned trajectory and ground truth trajectory is used as the main metric. Additionally, the Collision Rate (CR) is evaluated. PEP achieves the lowest L2 distance at the last planned position, $3\,\text{s}$ into the future as well as averaged along the trajectory. Regarding the CR, PEP performs slightly worse than methods, which additionally use ground truth velocity and acceleration as input. However, the performance is similar to other methods that, like PEP, do not do so. The results suggest that route attraction becomes increasingly beneficial the longer the planning horizon gets. Compared to SOTA, the L2($3\,\text{s}$) is reduced by $28.1\,\%$ while the L2(Avg.) decreases by $3.6\,\%$. We assume that the L2(Avg.) and CR could be further reduced by incorporating map information under consideration of roto-translation equivariance. Map information should lead to more accurate interaction modeling, which increases prediction and, thus, planning performance. Including a map will therefore be the next step to further improve our lightweight map-less approach. The qualitative results in Fig. 3 showcase how PEP benefits from prediction, route, and multi-modality.

E. Prediction

Even though prediction is not the primary task of PEP, it leverages joint prediction for realistic interaction modeling when planning the EV trajectory. During our planning experiments, we measured an SV prediction performance with a minL2(Avg.) of $0.82\,\text{m}$ and a minL2($3\,\text{s}$) of $0.99\,\text{m}$. Considering that no map is available for the SVs, these results are worth mentioning. In the following, we investigate whether planning really benefits from joint prediction.

F. Ablation

We present ablation studies for the major design choices of our model. To assess the impact of SV predictions on
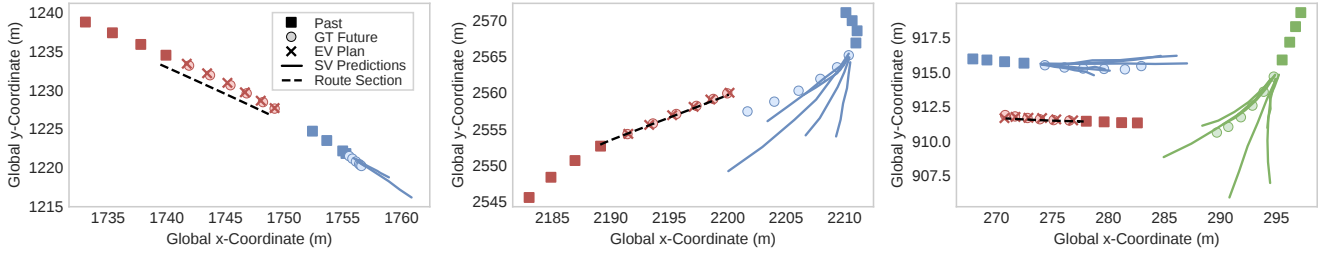
Fig. 3: Qualitative results. While the EV (red) uses the route (dashed) for guidance, it does not stick to it (left). Predicting actions of SVs improves EV planning, for example by anticipating SVs to decelerate (blue, left) or to cross the EV lane (blue, middle). Multi-modal predictions help the planner to consider diverse future scenarios (green, right).

TABLE II: Ablations of PEP model

Prediction	Route	Equivariance	L2 (m)		CR (%)	
			3 s	Avg.	3 s	Avg.
-	-	-	4.94	2.88	1.33	1.79
✓	✓	-	2.81	2.24	1.73	1.23
✓	-	✓	1.71	1.46	1.40	0.85
-	✓	✓	0.35	0.31	0.48	0.42
✓	✓	✓	0.32	0.28	0.43	0.37

EV planning performance, $\mathcal{L}_{\text{pred}}$ is removed from the loss function (c.f. Eq. 13) so that the model is not explicitly trained to predict SVs. Route ablation is realized by skipping the route attraction module described in Eq. 3. Finally, we deliberately destroy the SE(2)-equivariance of PEP by not subtracting and re-adding the mean position $\bar{\mathbb{X}}$ during equivariant feature initialization (c.f. Eq. 1). All networks are trained until convergence.

Overall, the ablation experiments show that each component contributes to the planning performance. Ablating all components at once yields the highest L2 distances and CR(Avg.) but not the highest CR(3 s). This is explainable by poor behaviors like driving off-road or stopping, which are the consequence of a map-less and route-less approach without explicit prediction. Such behaviors increase the L2 distance and reduce the CR in an unreasonable way. Next, we investigate the effect of ablating individual components. Ablating equivariance results in the highest L2(3 s) and L2(Avg.) increase, which indicates that the model benefits from the prior knowledge on scene symmetry that is integrated by means of SE(2)-equivariance. Not integrating this knowledge into the model architecture means that the model has to learn it itself, which reduces the sample efficiency and requires model capacity. Discarding the route also leads to a severe performance decrease as it takes away the only available map information, making the model fully interaction-based. In contrast, PEP performs only marginally worse when ablating explicit prediction, which is consistent with recent findings that EV information is decisive for open loop planning on nuScenes where interactions play a minor role [15], [52]. Our results show that prediction is less important than route information and equivariant model design. Nevertheless, ablating prediction leads to $-10.7\,\%$ L2(Avg.) and $-9.4\,\%$ L2(3 s) compared to the complete model.
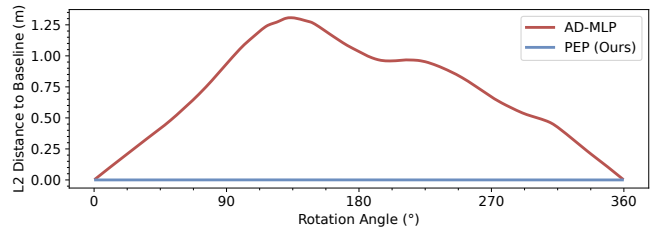

Fig. 4: Output stability. Inferred outside the training distribution, our SE(2)-equivariant model guarantees a stable output.

G. Equivariance

To investigate equivariance, we measure the output stability under input transformations. To this, the input trajectory and route are rotated by $\theta \in [1°, 2°, \ldots, 359°]$. Then, the EV trajectory is planned and transformed back into the baseline coordinate system by a rotation of $-\theta$. The trajectory planned without applying any rotation, i.e. $\theta = 0$, serves as the baseline. For an ideal equivariant model, the L2 distance to the baseline should be zero for all θ.

Fig. 4 depicts the output stability under rotation. Except for negligible numerical effects from rotation, the L2-distance is constant around zero, demonstrating that PEP is rotation-equivariant. Repeating the experiment with added random 2D translations confirms the results. In contrast, AD-MLP [15] which is trained on EV-centered data, is sensitive to input rotations which could, for example, arise from measurement errors. Especially when designing safety-relevant systems for automated driving, output stability and explainable behavior under input transformations are crucial.

V. CONCLUSION

In this work, we have proposed PEP, a simplistic equivariant planning model that integrates prediction and planning in a joint approach. Our experiments show that PEP achieves state-of-the-art performance in open-loop planning on nuScenes. Three major design choices contribute to the performance: Joint prediction and planning, our novel route attraction module, and the SE(2)-equivariant network design. We demonstrate output stability under transformations of the input. This property of equivariant models can provide safety guarantees and might become an important aspect in the future of automated driving.

REFERENCES

[1] M. Hallgarten, M. Stoll, and A. Zell, "From prediction to planning with goal conditioned lane graph traversals," *arXiv:2302.07753*, 2023.

[2] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proc. of IEEE/CVF ICCV*, 2019, pp. 2821–2830.

[3] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen, "Pip: Planning-informed trajectory prediction for autonomous driving," in *Computer Vision–ECCV 2020, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 598–614.

[4] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proc. of IEEE/CVF CVPR*, 2023, pp. 17 853–17 862.

[5] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," *arXiv:2303.05760*, 2023.

[6] S. Hagedorn, M. Hallgarten, M. Stoll, and A. Condurache, "Rethinking integration of prediction and planning in deep learning-based automated driving systems: a review," *arXiv:2308.05731*, 2023.

[7] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun, "Gorela: Go relative for viewpoint-invariant motion forecasting," in *2023 IEEE ICRA*. IEEE, 2023, pp. 7801–7807.

[8] V. G. Satorras, E. Hoogeboom, and M. Welling, "E (n) equivariant graph neural networks," in *ICML*. PMLR, 2021, pp. 9323–9332.

[9] C. Xu, R. T. Tan, Y. Tan, S. Chen, Y. G. Wang, X. Wang, and Y. Wang, "Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning," in *IEEE/CVF CVPR*, 2023, pp. 1410–1420.

[10] R. Walters, J. Li, and R. Yu, "Trajectory prediction using equivariant continuous convolution," *arXiv:2010.11344*, 2020.

[11] M. Rath and A. P. Condurache, "Improving the sample-complexity of deep classification networks with invariant integration," *arXiv:2202.03967*, 2022.

[12] ——, "Boosting deep neural networks with geometrical prior knowledge: A survey," *arXiv:2006.16867*, 2020.

[13] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *ICLR*, 2021.

[14] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Thomas: Trajectory heatmap output with learned multi-agent sampling," *arXiv:2110.06607*, 2021.

[15] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes," *arXiv:2305.10430*, 2023.

[16] M. Hallgarten, I. Kisa, M. Stoll, and A. Zell, "Stay on track: A frenet wrapper to overcome off-road trajectories in vehicle motion prediction," *arXiv:2306.00605*, 2023.

[17] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *IEEE/CVF CVPR*, 2020, pp. 11 525–11 533.

[18] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 ICRA*. IEEE, 2019, pp. 2090–2096.

[19] F. Janjoš, M. Dolgov, and J. M. Zöllner, "Starnet: Joint action-space prediction with star graphs and implicit global-frame self-attention," in *2022 IEEE IV*. IEEE, 2022, pp. 280–286.

[20] J. Kim, R. Mahjourian, S. Ettinger, M. Bansal, B. White, B. Sapp, and D. Anguelov, "Stopnet: Scalable trajectory and occupancy prediction for urban autonomous driving," in *IEEE ICRA*, 2022, pp. 8957–8963.

[21] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, "3d steerable cnns: Learning rotationally equivariant features in volumetric data," *Advances in NeuIPS*, vol. 31, 2018.

[22] M. Weiler and G. Cesa, "General e (2)-equivariant steerable cnns," *Advances in neural information processing systems*, vol. 32, 2019.

[23] Y. Wang and J. Chen, "Eqdrive: Efficient equivariant motion forecasting with multi-modality for autonomous driving," *arXiv:2310.17540*, 2023.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[25] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proc. of IEEE/CVF CVPR*, 2020, pp. 11 621–11 631.

[26] A. Cui, S. Casas, A. Sadat, R. Liao, and R. Urtasun, "Lookout: Diverse multi-future prediction and planning for self-driving," in *Proc. of IEEE/CVF ICCV*, 2021, pp. 16 107–16 116.

[27] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," *arXiv:2306.07962*, 2023.

[28] Y. Chen, P. Karkus, B. Ivanovic, X. Weng, and M. Pavone, "Tree-structured policy planning with learned behavior models," *arXiv:2301.11902*, 2023.

[29] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko, "Safe real-world autonomous driving by learning to predict and plan with a mixture of experts," *arXiv:2211.02131*, 2022.

[30] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *arXiv:2207.10422*, 2022.

[31] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *ICML*. PMLR, 2016, pp. 2990–2999.

[32] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, "Rotation equivariant vector field networks," in *IEEE ICCV*, 2017, pp. 5048–5057.

[33] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar transformer networks," *arXiv:1709.01889*, 2017.

[34] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. of IEEE CVPR*, 2017, pp. 5028–5037.

[35] M. Weiler, F. A. Hamprecht, and M. Storath, "Learning steerable filters for rotation equivariant cnns," in *IEEE CVPR*, 2018, pp. 849–858.

[36] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds," *arXiv:1802.08219*, 2018.

[37] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "Se (3)-transformers: 3d roto-translation equivariant attention networks," *Advances in neural information processing systems*, vol. 33, pp. 1970–1981, 2020.

[38] B. Jing, S. Eismann, P. Suriana, R. J. Townshend, and R. Dror, "Learning from protein structure with geometric vector perceptrons," *arXiv:2009.01411*, 2020.

[39] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas, "Vector neurons: A general framework for so (3)-equivariant networks," in *Proc. of IEEE/CVF ICCV*, 2021, pp. 12 200–12 209.

[40] M. Kofinas, N. Nagaraja, and E. Gavves, "Roto-translated local coordinate frames for interacting dynamical systems," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6417–6429, 2021.

[41] B. Ummenhofer, L. Prantl, N. Thuerey, and V. Koltun, "Lagrangian fluid simulation with continuous convolutions," in *ICLR*, 2019.

[42] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *ICML*. PMLR, 2020, pp. 8459–8468.

[43] R. Wang, R. Walters, and R. Yu, "Incorporating symmetry into deep dynamics models for improved generalization," *arXiv:2002.03061*, 2020.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[45] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proc. of IEEE/CVF CVPR*, 2019, pp. 8660–8669.

[46] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *Proc. of IEEE/CVF CVPR*, 2021, pp. 12 732–12 741.

[47] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, and D. Ramanan, "Differentiable raycasting for self-supervised occupancy forecasting," in *ECCV*. Springer, 2022, pp. 353–369.

[48] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *ECCV*. Springer, 2022, pp. 533–549.

[49] Z. Chen, M. Ye, S. Xu, T. Cao, and Q. Chen, "Deepemplanner: An em motion planner with iterative interactions," *arXiv:2311.08100*, 2023.

[50] T. Ye, W. Jing, C. Hu, S. Huang, L. Gao, F. Li, J. Wang, K. Guo, W. Xiao, W. Mao, *et al.*, "Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving," *arXiv:2308.01006*, 2023.

[51] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," *arXiv:2303.12077*, 2023.

[52] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?" *arXiv:2312.03031*, 2023.