

Fig. 2. Parametric representations of the scenarios (a) adaptive cruise control (ACC, “The Ego follows the Co on a straight road.”), (b) lane keeping (LK, “The Ego follows a curved road.”), and (c) ACC&LK (“The Ego follows the Co on a curved road.”). The center of the lanes in (b) and (c) follows a polynomial of 3rd degree. In (a) and (c), the Co starts from its initial x -coordinate x_{Co} with a velocity of v_{Co} . After a time of $t_{v_{Co}}$, it decelerates with a_{Co} for $t_{a_{Co}}$. The Ego’s initial velocity v_{Ego} is varied in all scenarios. More details about the variation are given in [Section IV-A](#). As a result of the execution, evaluation metrics are calculated: the Ego’s minimum deceleration a_{min} , the Ego’s maximum lateral position p_{lat_max} , and the vehicles’ minimum distance d_{min} .

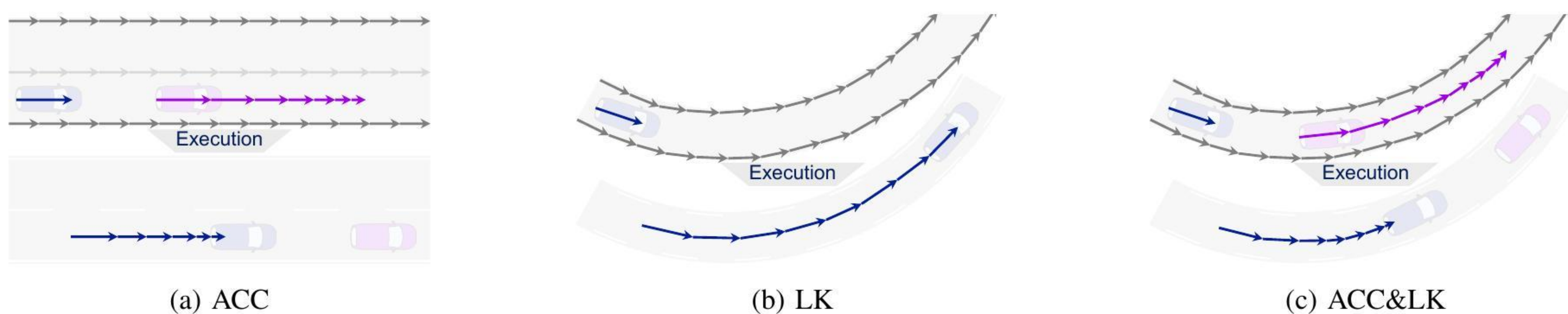


Fig. 3. Vectorized representations of the above scenarios. The scenarios are defined by the static environment (lanes, gray), the dynamic environment (Co, purple), and the Ego’s initial pose and velocity (blue). The execution determines the trajectory of the Ego, which can be utilized to calculate evaluation metrics as the ones in [Fig. 2](#). To account for interactions between the Ego and Co(s), trajectories of Co(s) may also result from execution (see [Section VI](#)).

sensor noise). Considering scenarios’ elements [\[21\]](#), such a translation works for environmental conditions, which parameters can characterize. Digital information is specific to the AV under test and will mostly not change. However, the road network and traffic guidance objects (layer 1 (L1)), roadside structures (L2), temporary modifications of L1 and L2 (L3), and dynamic objects (L4) cannot be described well by parameters. Hence, translating between concrete scenarios’ representations might not be possible for changes in L1 to L4.

3) *Transfer Across Functional Scenarios*: A change of the functional scenario mostly leads to changes in L1 to L4 and evaluation metrics. Therefore, a transfer of data can not be handled by the discussed search-based techniques.

III. INTEGRATING SCENARIO-BASED TESTING AND MOTION PREDICTION

This section illustrates the previously described problems with an example and presents an approach to solving them. Considering the three functional scenarios in [Fig. 2](#), it is desirable that data from the scenarios adaptive cruise control (ACC) and lane keeping (LK) allow to predict the scenario ACC&LK’s outcomes. However, the different logical scenarios and evaluation metrics stand against such a transfer. To enable the transfer of data across different functional (and hence logical) scenarios, metamodels must get more information than just concrete scenarios and evaluation metrics. To implement this, we propose scenario *embeddings*.

A. From Concrete Scenarios to Scenario Embeddings

An embedding can be defined as “a relatively low-dimensional space into which you can translate high-

dimensional vectors” [\[22\]](#). Since the instantiated scenario cannot be described by structured data, we create scenario embeddings holding information added during the transition from concrete to instantiated scenarios. Thereby, scenario embeddings can be created without (costly) scenario execution.

A scenario embedding can have various representations. To cover changes in L1 to L4, rendered or vectorized representations from motion prediction are particularly suitable [\[23\]](#) (see upper part of [Fig. 3](#)). In [\[6\]–\[8\]](#), such representations are used to generate synthetic scenarios based on real scenarios. However, the behavior of an Ego and the resulting criticality are not considered. Our approach is thus complementary, enabling predictive analyses for synthetic scenarios as well.

Since vectorized representations are characterized by a high level of detail and computational efficiency [\[23\]](#), we choose a vectorized scenario description for our scenario embeddings.

B. From Evaluation Metrics to Trajectories

To allow for the transfer of data across different evaluation metrics, scenarios’ outcomes must be captured in a uniform representation independent of the utilized evaluation metrics. To assess the criticality of scenarios, most evaluation metrics are calculated based on the trajectories of the Ego and other objects. Since the static and dynamic environment of the Ego is already modeled in the scenario embedding, we consider the Ego’s trajectory the product of the execution (see lower part of [Fig. 3](#)), allowing for the calculation of evaluation metrics.

C. Motion Prediction for Scenario-Based Testing

[Fig. 1](#) shows that predictive analyses require trajectories to be predicted based on scenario embeddings. A model suitable