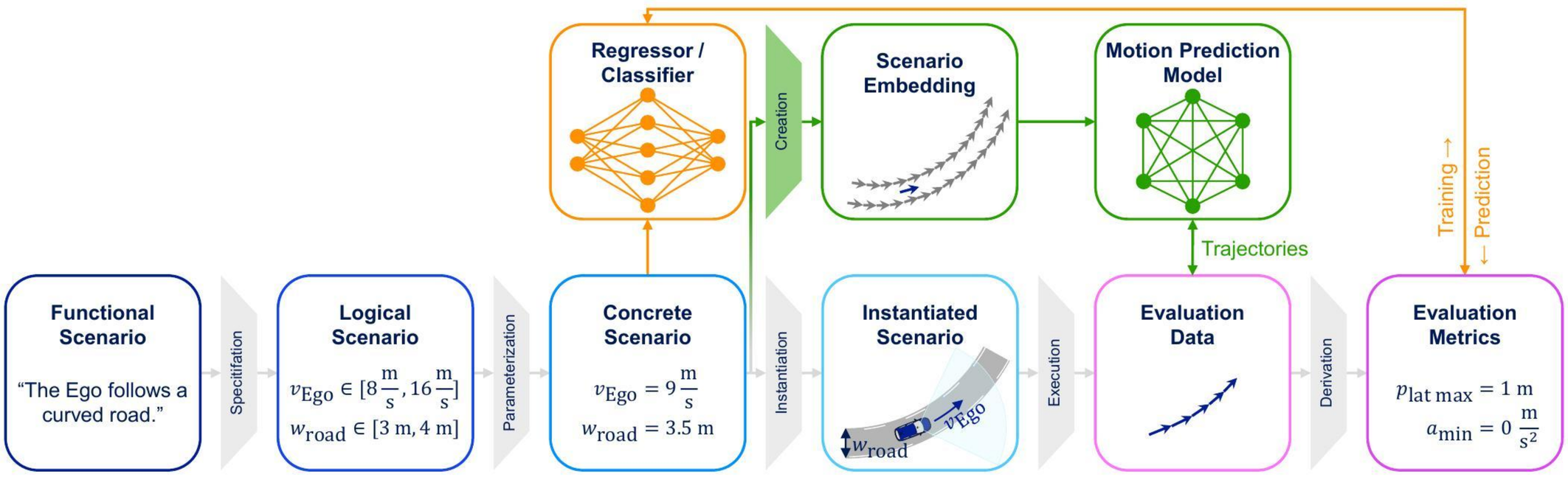

Fig. 1. Artifacts in scenario-based testing. While the literature primarily considers the successive definition of functional, logical, and concrete scenarios, tests ultimately have to be instantiated in a test setup. Thus, we refer to the scenario that is executed as the *instantiated* scenario. During execution, evaluation data are generated, which usually include objects' trajectories and log data. Based on these, evaluation metrics are derived to enable automated assessment. Section II-A to II-C explain these artifacts and transitions in detail. To generate relevant concrete scenarios, current search-based techniques learn a mapping from concrete scenarios to the evaluation metrics, e.g., via regression or classification models (see Section II-D). To learn and predict across functional scenarios and evaluation metrics, we propose using motion prediction models that predict objects' trajectories based on scenario embeddings (see Section III).

Concrete scenarios assign fixed values to the inputs of a logical scenario [5]. Hence, each concrete scenario is a parameterized sample $\boldsymbol{x}$ that can be processed automatically.

B. From Concrete Scenarios to Instantiated Scenarios

Typically, concrete scenarios are considered the final representation before execution [5]. However, to execute concrete scenarios, they need to be implemented in a test setup [11], [12]. Thereby, some essential properties are determined, which may not be defined in standardized formats such as OpenScenario, e.g., material properties of objects. Since such properties may significantly influence the outcomes of concrete scenarios' execution, we define a more concrete type of scenario:

Instantiated scenarios are scenarios that are ready to be executed in a test setup and include everything that is determined prior to execution. For virtual test setups, the instantiated scenario includes the environment models and their configuration. On a proving ground or during field tests, only some properties of the instantiated scenario are controllable, while others may not be controllable or are even unknown. Since the instantiated scenario can take on various digital, physical, or mixed forms, it cannot be preserved or processed in a uniform data representation.

C. From Instantiated Scenarios to Evaluation Metrics

Two artifacts result from executing instantiated scenarios:

Evaluation data are generated during instantiated scenarios' execution [11] and typically include objects' trajectories and log data. Some data can only be recorded in virtual test setups, e.g., ground truth object positions.

Evaluation metrics are derived from evaluation data [11] and chosen based on the test criteria (e.g., comfort or safety), the functional scenario investigated, and the test setup used. E.g., purely longitudinal scenarios may use a minimum time to collision, and more complex scenarios a post encroachment time. Analogous to concrete scenarios, evaluation metrics are outputs, and N_O evaluation metrics are a sample $\boldsymbol{y} \in \mathbb{R}^{N_O}$.

D. Search-Based Techniques for Scenario Selection

Since relevant concrete scenarios are rare and their execution can be resource-intensive, search-based techniques are used to systematically parameterize scenarios [4], [12]–[15], e.g., via Bayesian optimization, genetic algorithms, reinforcement learning, or adaptive importance sampling. These methods commonly treat the transition from a concrete scenario $\boldsymbol{x}$ to evaluation metrics $\boldsymbol{y}$ as a black-box [12]. This is most apparent for regression and classification metamodels, which approximate the mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ based on N_S samples forming a dataset $\boldsymbol{X} \in \mathbb{R}^{N_S \times N_I}$, $\boldsymbol{Y} \in \mathbb{R}^{N_S \times N_O}$ [16], [17]. This way, metamodels can bypass the resource-intensive execution by predictions (see upper part of Fig. 1).

Search-based techniques can significantly increase the efficiency of scenario-based testing [13]–[17]. However, the sole reliance on concrete scenarios and evaluation metrics is restrictive: For different logical scenarios (more, less, or different inputs), the nature of $\boldsymbol{x}$ changes and data cannot simply be compared or combined; the same is true for evaluation metrics and $\boldsymbol{y}$. If the transition between $\boldsymbol{x}$ and $\boldsymbol{y}$ changes, $\boldsymbol{x}$ and $\boldsymbol{y}$ have the same nature but may have a very different relationship. Thus, here too, data cannot always be compared or combined.

Due to the described problems, most search-based techniques can only be used for a fixed logical scenario with fixed evaluation metrics. However, considering that many functional and logical scenarios with different evaluation metrics have to be analyzed, more extensive transfers of data are beneficial:

1) Transfer Across Instantiated Scenarios: [18] and [19] consider that multiple test setups are available for a given logical scenario and evaluation metric, which allows the advantages of different test setups to be combined. However, as discussed in Section II-A, a change of the test setup may allow for or require a change of the logical scenario.

2) Transfer Across Logical Scenarios: In [20], changes of the logical scenario are handled by translating between concrete scenarios with different inputs (e.g., from rain to