Learning to Predict Vehicle Trajectories with Model-based Planning

Haoran Song, Di Luan, Wenchao Ding, Michael Yu Wang, Qifeng Chen
Hong Kong University of Science and Technology

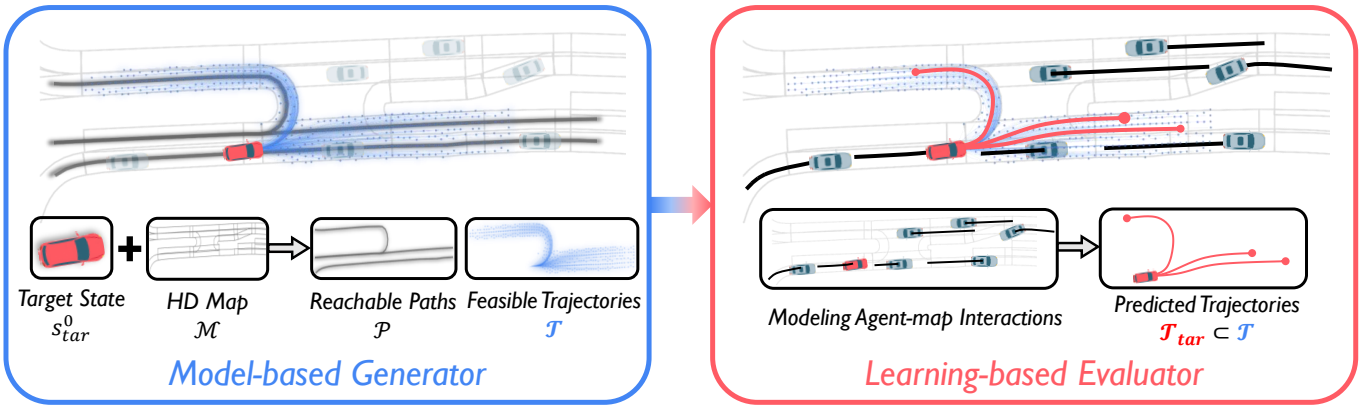

Fig. 1. Illustration of our proposed PRIME framework. PRIME has two stages for trajectory prediction in traffic scenarios: the model-based generator (left) which samples the target's feasible future trajectories $\mathcal{T}$ by taking its real-time state $\mathbf{s}^0_{tar}$ and the map $\mathcal{M}$, while explicitly imposing kinematical and environmental constraints to guarantee trajectory feasibility; the learning-based evaluator (right) which receives the feasible trajectories $\mathcal{T}$ and all observed tracks $\mathcal{S}$ to model the implicit interactions among all traffic agents, and selects a final set of feasible trajectories $\mathcal{T}_{tar} \subset \mathcal{T}$ as the prediction result.

***Abstract*—Predicting the future trajectories of on-road vehicles is critical for autonomous driving. In this paper, we introduce a novel prediction framework called PRIME, which stands for Prediction with Model-based Planning. Unlike recent prediction works that utilize neural networks to model scene context and produce unconstrained trajectories, PRIME is designed to generate accurate and feasibility-guaranteed future trajectory predictions, which guarantees the trajectory feasibility by exploiting a model-based generator to produce future trajectories under explicit constraints and enables accurate multimodal prediction by using a learning-based evaluator to select future trajectories. We conduct experiments on the large-scale Argoverse Motion Forecasting Benchmark. Our PRIME outperforms state-of-the-art methods in prediction accuracy, feasibility, and robustness under imperfect tracking. Furthermore, we achieve the 1st place on the Argoervese Leaderboard.** [1]

I. Introduction

Predicting the future states of dynamic agents is critical for robot planning in an interactive environment. In the architecture of autonomous driving, prediction serves as the bridging module that reasons future states based on the perceived information from upstream detection and tracking and provides the predicted future states to facilitate the downstream planning. Therefore, we are particularly interested in making accurate and reasonable trajectory predictions for on-road vehicles, which is vital for planning safe, efficient, and comfortable motion for self-driving vehicles (SDVs).

[1]Our approach ranks the 1st on the Argoervese Motion Forecasting Leaderboard by 2021-03-01. The snapshot is shown in the appendix (Fig. 7)

Trajectory prediction is challenging because the self-driving system has various requirements for this module. The widely known difficulty lies in modeling multi-agent interactions in an on-road environment and inferring multimodal future states. Traditional methods [16, 18, 44, 21] produce motion forecasting by handcrafted rules or models with embedded physical and environmental features, which are insufficient for modeling interactive agents in complex scenes. In recent years, learning-based approaches [1, 20, 2] advance the frontier of this area. With deep neural networks to fuse scene context information and generate future trajectories, learning-based frameworks significantly promote the prediction accuracy and dominate the recent motion forecasting competitions for autonomous driving [3, 7].

Despite achieving steady improvement in accuracy, much less attention has been paid to the feasibility and robustness in prediction. Indeed, most traffic participants operate under their inherent kinematic constraints (e.g., non-holonomic motion constraints for vehicles) while in compliance with the road structure (e.g., lane connectivity, static obstacles) and semantic information (e.g., traffic lights, speed limits). All these kinematic and environmental constraints explicitly regularize the trajectory space. However, most existing future prediction methods model traffic agents as points and produce sequences of future positions without constraints. Such constraint-free predictions may be incompliant with kinematic or environmental characteristics, which gives rise to massive uncertainty in the predicted future states. Consequently, the downstream

planning module would inevitably undergo some extra burdens, and even the "freezing robot problem" [37]. Furthermore, recent works typically generate trajectory predictions by network regression, which has high dependences on long-term tracking results. Nevertheless, for some dense driving scenarios, where the target would be momently occluded or suddenly appears within the sensing range, tracking results are discontinuous or not accumulated enough. The prediction accuracy would degrade under such imperfect tracking cases.

Toward overcoming these challenges, we propose a novel prediction architecture called PRIME. The critical idea is to exploit a model-based motion planner as the prediction **generator** to sample feasible future trajectories for targets, together with a deep neural network as the prediction **evaluator** to select future trajectories by scoring. The novel architecture contributes to accurate, feasible, and robust trajectory predictions. Our contributions are summarized as follows:

- We propose PRIME, a novel framework for vehicle trajectory prediction. PRIME guarantees the feasibility of trajectory predictions by exploiting a model-based generator to produce future trajectories under explicit constraints, while it enables accurate multimodal prediction by using a learning-based evaluator to select future trajectories.
- Our PRIME framework ranks the 1st on the Argoverse Motion Forecasting Leaderboard, reaching the best performance on Miss Rate (MR_6, official ranking metric) and Probabilistic Final Displacement Error (p-$minFDE_6$). Notably, to the best of our knowledge, PRIME is the only method that uses an interpretable motion planner to produce trajectory prediction among the top-ranked entries.
- Our PRIME outperforms the state-of-the-art methods in trajectory feasibility and prediction robustness under imperfect tracking in addition to the prediction accuracy. These attributes facilitate more efficient downstream planning for self-driving vehicles.

II. Related Work

This section first reviews the typical motion prediction and planning methods in self-driving, focusing on their connection and difference. We then cover the recent advances in modeling agent-map interactions and generating multimodal predictions, which are fundamental issues in the prediction task.

Prediction and Planning are closely intertwined in the pipeline of autonomous driving [12, 39, 19, 34]. Planning is predominantly studied to generate kinematically feasible and environmentally compatible trajectories and, with considering more aspects such as comfort, safety, energy consumption, and progress towards the goal, select the best plan for the self-driving vehicle (ego agent). Prediction facilitates the best plan selection by inferring future trajectories of the surrounding vehicles (target agents) from the perceived historical information. Their different primary focuses make the corresponding mainstream frameworks diverge. Model-based approaches [27, 40, 24, 13] are preferred in planning due to their interpretability and reliability in computing safe trajectories under explicit constraints. Learning-based methods [25], in contrast, prevail in prediction by utilizing the power of data-driven in modeling implicit multi-agent interactions.

Some learning-based prediction methods incorporate the goal-directed idea from planning to infer the possible goals and then produce goal-conditioned trajectories using inverse reinforcement learning (IRL) [43, 29] or deep neural networks [23, 41]. Moreover, several recent works introduce novel planning-prediction-coupled frameworks to perform conditional prediction with respect to ego's intentions [31] and motion plans [35, 33]. With much emphasis on improving the point-level prediction accuracy, all these learning-based works rely on neural networks to process complex traffic environments, but cannot ensure physical constraints are indeed imposed on trajectory generation. With the exception of DKM [9], it embeds the two-axle vehicle kinematics [28] in the output layer to ensure kinematically feasible trajectories, yet still no guarantee on environmental feasibility.

Inspired by the popular sampling-based paradigm in vehicle motion planning [39, 40], which samples abundant trajectories subject to explicit model constraints and then, based on predefined scoring functions, chooses the best trajectory for execution. We employ a model-based motion planner in trajectory generation to provide trajectory sets based on the real-time state of prediction targets and road environments. Afterward, a learning-based network's role is simplified to rank the set of feasible trajectories with modeling agent-map interactions. In this way, our novel two-stage architecture makes the most of model-based planning and learning-based prediction, which fulfills environmental and kinematic constraints while handling complex interactions.

Modelling agent-map interactions is critical for prediction. The classical work from Benz [44] anticipates driving behavior with map constraints. It first associates each target with corresponding reachable lanes and then directly generates lane-following trajectory predictions based on the target's state and map topology. However, it fails in capturing the multi-agent interaction that broadly exists in interactive driving scenarios. To better capture information from road environment and dynamic agents, many learning-based works [8, 11, 26] convert raw input data to rasterization images by rendering traffic entities with different colors or intensities, which could be encoded with Convolutional Neural Networks. More recent works [14, 22, 41] propose to use vectorized scene context as nodes to construct a graph, followed by processing with Graph Neural Networks. The vectorized representation exploits High Definition (HD) maps more explicitly and brings improvement in prediction accuracy. Different from these approaches, we address the agent-map modeling through a hierarchical structure that incorporates the lane-association ideas from [44] while extends to learn global scene context. To be specific, our prediction generator acts locally in a planning manner to generate trajectory sets for the target vehicle on reachable paths. Next, our prediction evaluator gets a global understanding of the scene context by learning from aggregating over all lane-

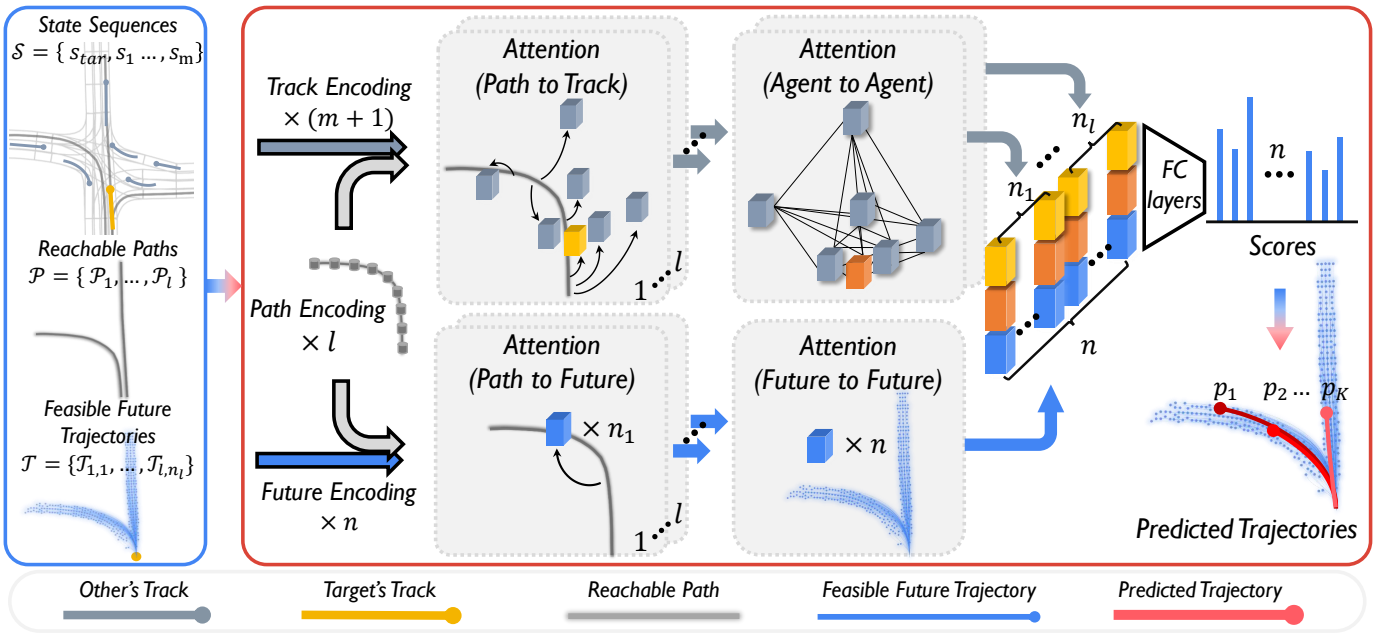

Fig. 2. PRIME framework overview. The model-based generator searches reachable paths $\mathcal{P}$ through the map and samples a set of feasible future trajectories $\mathcal{T}$. The learning-based evaluator first encodes scene context given by $(\mathcal{P}, \mathcal{T}, \mathcal{S})$, including l paths in $\mathcal{P}$, $(m+1)$ history tracks in $\mathcal{S}$ and n future trajectories in $\mathcal{T}$. The implicit agent-map interactions are learned in the subsequent attention modules: P2T and P2F propagate the spatial information of each reference path $\mathcal{P}_i$ into history tracks and corresponding future trajectories, and A2A takes track tensors from P2T to capture the multi-agent interactions. As the path-based Frenét coordinate is used in our dual spatial representation, P2T, P2F, and A2A operate for each path, while F2F fuses all the future trajectories processed by P2F to obtain a global understanding for the reachable space. Subsequently, each feasible trajectory $\mathcal{T}_{i,j}$ could query its track tensor $\mathbf{X}_i(\mathbf{s}_{tar})$ from P2T, interaction tensor $\mathbf{Y}_i(\mathbf{s}_{tar})$ from A2A and future tensor $\mathbf{Z}(\mathcal{T}_{i,j})$ from F2F, and it is scored by feeding the concatenation of these tensors to fully-connected layers. Finally, the evaluator ranks all feasible future trajectories in $\mathcal{T}$ by scoring and outputs a final set of K predicted trajectories.

conditioned trajectories and map features.

Generating multimodal trajectories is another core challenge for prediction. To account for the intrinsic multimodal distributions, a line of works are built upon stochastic models such as conditional variational autoencoders (CVAEs) [20, 30, 17, 36, 5] or generative adversarial networks (GANs) [15, 32, 42] to draw trajectory samples. Although their performance is competitive, the drawback of sampling uncontrollable latent variables at inference time prohibits them from being deployed on safety-critical driving scenarios. Deterministic approaches are mostly based on multi-mode trajectory regression [10, 4, 8, 22]. To alleviate mode collapse in prediction learning, recently proposed frameworks decompose the task into classification over anchor trajectories [6] or goal-conditioned trajectories [41], followed by trajectory offset regression. However, there is no feasibility guarantee for the trajectory regressed from neural networks. CoverNet [26] attempts to meet specific physical requirements by formulating multimodal prediction as pure classification over a pre-constructed trajectory set, but its prediction may violate the real-time states of agents or traffic environment. By leveraging model-based planning as the prediction generator, our approach shows superiority in the following crucial aspects. Firstly, trajectory feasibility could be guaranteed by imposing environmental and kinematic constraints from real-time situations. Secondly, our model provides multimodal distribution by generating trajectory sets with sufficient coverage over reachable paths. Thirdly, rather than outputs discrete future locations as most methods do, our model produces high-fidelity trajectories with continuous information, including location, heading, velocity, acceleration, etc. Lastly, it could generate trajectories given the target's current state, which mitigates the high dependence on long-term tracking results.

III. Overview

Problem formulation. We assume that the self-driving vehicle is equipped with detection and tracking modules that provide observed state $\mathcal{S}$ for on-road agents $\mathcal{A}$, and has access to the HD map information $\mathcal{M}$. Let $\mathbf{s}_i^t$ denote the state of agent $a_i \in \mathcal{A}$ at frame t, including position, heading, velocity, turning rate and actor type, and $\mathbf{s}_i = \left\{\mathbf{s}_i^{-T_P+1}, \mathbf{s}_i^{-T_P+2}, ..., \mathbf{s}_i^0\right\}$ denote the sequence of discrete states throughout the observed period T_P. Given any agent as the prediction target, we denote it by $\mathbf{a}_{tar}$ and the other surrounding agents as $\mathcal{A}_{nbrs} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_m\}$ for differentiation. Then we correspondingly denote their state sequences as $\mathbf{s}_{tar}$ and $\mathcal{S}_{nbrs} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m\}$. Accordingly, we have $\mathcal{S} = \{\mathbf{s}_{tar}\} \cup \mathcal{S}_{nbrs}$ and $\mathcal{A} = \{\mathbf{a}_{tar}\} \cup \mathcal{A}_{nbrs}$. The objective of our framework is to predict multimodal future trajectories $\mathcal{T}_{tar} = \{\mathcal{T}_i | i = 1, 2, ..., K\}$ together with corresponding trajectory probability $\{p_i | i = 1, 2, ..., K\}$, where $\mathcal{T}_i$ denotes a predicted trajectory for target agent $\mathbf{a}_{tar}$ with continuous state information up to the prediction horizon T_F, K is the number of modes. Furthermore, it is required to guarantee each trajectory $\mathcal{T}_i \in \mathcal{T}_{tar}$ are feasible with respect to the existing constrains $\mathcal{C}$, which includes environmental constrains $\mathcal{C}_{\mathcal{M}}$ and target's kinematic constrains $\mathcal{C}_{tar}$.

Our framework. We tackle the trajectory prediction problem by introducing a two-stage architecture consisting of model-based generator G and learning-based evaluator E. The schematic illustration is shown in Fig. 1. Concretely, the generator $G : (\mathbf{s}^0_{tar}, \mathcal{M}, \mathcal{C}) \mapsto (\mathcal{P}, \mathcal{T})$ is tasked to produce the real-time trajectory space $\mathcal{T}$ for the target agent, which is approximated by a finite number of feasible trajectories. This part starts with searching a set of reachable paths $\mathcal{P} = \{\mathcal{P}_i | i = 1, 2, ..., l\}$ from the map information given by $\mathcal{M}$, which provides reference paths for trajectory generation. Then a classical sampling-based planner is utilized to generate trajectories samples under explicit constraints in $\mathcal{C}$, and thereby a set of feasible future trajectories $\mathcal{T} = \bigcup_{i=1}^{l} \{\mathcal{T}_{i,j} | j = 1, 2, ..., n_i\}$ is produced for the prediction target, in which $\mathcal{T}_{i,j}$ denotes the j-th feasible trajectory generated from $\mathcal{P}_i$, and the total number of trajectories $n = \sum_{i=1}^{l} n_i$.

In short, G is specialized in trajectory generation with feasibility guarantee but ignores interaction with surrounding agents $\mathcal{A}_{nbrs}$. While for the evaluator $E : (\mathcal{P}, \mathcal{T}, \mathcal{S}) \mapsto (\mathcal{T}_{tar}, \{p_i\})$, it takes charge of modeling implicit multi-agent interaction and accordingly selects the most probable future trajectories $\mathcal{T}_{tar}$ from $\mathcal{T}$. Note that the numbers for reachable paths l, surrounding agents m, and feasible trajectories n are varying with the real-time state of ($\mathcal{M}$, $\mathcal{S}$). Therefore, we implement E by training a deep neural network with attention mechanisms for dealing with dynamic numbers. Notably, the evaluator E is only tasked to score feasible trajectories in $\mathcal{T}$ without the need for regressing position or displacement as most learning-based frameworks do. The following two sections describe G and E in detail.

IV. Model-based Generator

A. Path Search

Unlike motion planning, where the reference path for the controllable ego is given, the prediction has no access to the future paths of uncontrollable targets. Therefore, we conduct path search G_{path} in advance of trajectory generation G_{traj}, such that each prediction target could be associated with a set of potential paths $\mathcal{P}^+$. Also, the relatively short time horizon (less than 5 seconds) in prediction makes it practicable to search all the potential paths for any on-road vehicles.

Our path search algorithm G_{path} is partially built upon the baselines proposed in [1]. Firstly we localize a_{tar} on the map and query its surrounding lane segments as the root segments. With the lane connectivity information provided by HD map $\mathcal{M}$, we search the segment sequences along the predecessors and successors of each root via Depth-First-Search on $\mathcal{M}$, up to forward distance D_F and backward distance D_B. Following, we concatenate each pair of forward and backward segment sequences and remove redundant ones with heuristics, and finally, the centerline coordinates of each segment sequence yield a potential path $\mathcal{P}_i \in \mathcal{P}^+$. In the phase of path search $G_{path} : (\mathcal{M}, \mathbf{s}^0_{tar}) \mapsto \mathcal{P}^+$, we expect the resulted path set $\mathcal{P}^+$ to provide sufficient coverage to the future path space of a_{tar}, but not impose dynamic constraints. So for target a_{tar} with current state $\mathbf{s}^0_{tar}$, some paths in $\mathcal{P}^+$ may not be reachable at frame $t = T_F$. For instance, a high-speed vehicle cannot change to the opposite lane with a U-turn in few seconds. Such unreachable paths could be recognized in the following trajectory generation part as no trajectories samples towards them are feasible. Then only the reachable paths would be reserved in $\mathcal{P} \subseteq \mathcal{P}^+$.

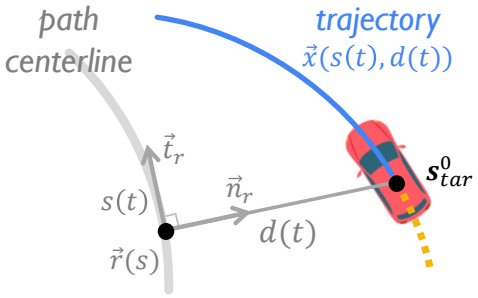

Fig. 3. Trajectory generation in a Frenét Frame

B. Trajectory Generation

Given the potential paths in $\mathcal{P}^+$ as dynamic references, we choose to generate future trajectories in a planning manner. In the driving domain, motion planning typically stands for finding an optimal trajectory for SDV to connect the current state and a goal state, which is essentially different from predicting multiple probable trajectories for vehicles with unknown intentions. Despite this, the model-based generator in planning, which computes a large number of trajectory samples for the follow-up selection, could also be exploited for the prediction.

To this end, we adopt the trajectory generation phase of Frenét planner [40] by Werling et al., which generates traffic-adapted trajectories in a Frenét Frame with fulfilling restrictions. The process of our trajectory generator $G_{traj} : (\mathcal{P}^+, \mathbf{s}^0_{tar}, \mathcal{C}) \mapsto \mathcal{T}$ is illustrated in Fig. 3. Given a refernece path in $\mathcal{P}^+$, a dynamic curvilinear frame is given by the tangential vector $\vec{t}_r$ and normal vector $\vec{n}_r$ at a certain point r on the path centerline. The Cartesian coordinate $\vec{x} = (x, y)$ could be converted to the Frenét coordinate (s, d) with the relation

$$\vec{x}(s(t), d(t)) = \vec{r}(s(t)) + d(t)\vec{n}_r(s(t)), \tag{1}$$

in which $\vec{r}$ represents a vector pointing from the path root, s and d denote the covered arc length and the perpendicular offset. The trajectory generation begins with projecting the current state $\mathbf{s}^0_{tar}$ onto the Frenét frame and obtains the state tuple $[s_0, \dot{s}_0, \ddot{s}_0, d_0, \dot{d}_0, \ddot{d}_0]$. The longitudinal movement $s(t)$ and lateral movement $d(t)$ within the prediction horizon T_F are then generated independently by connecting the fixed start state tuple with different end state tuples using polynomial curves to cover different driving maneuvers.

Compared with planning for a controllable agent, prediction receives less accurate state estimation for uncontrollable targets, and it does not need fine-grained trajectories. In our trajectory generation phase, therefore, some high-order state

variants are simplified to zero, including $\ddot{s}(0)$, $\ddot{d}(0)$ of the start state and $\ddot{s}(T_F)$, $\ddot{d}(T_F)$, $\dot{d}(T_F)$ of the end states. For longitudinal movement, we sample the target velocity $\dot{s}(T_F) \leftarrow \dot{s}_i$ in the range of $[\max(0, \dot{s}_0 - \delta^- T_F), \min(\hat{\dot{s}}, \dot{s}_0 + \delta^+ T_F)]$ while leave $s(T_F)$ unconstrained. The constants δ^-, δ^+ and $\hat{\dot{s}}$ are given by considering the actor type of $\mathbf{a}_{tar}$ and speed limit in $\mathcal{M}$, to control the sampled longitudinal velocity in a reasonable range. Each longitudinal trajectory $s_i(t)$ is calculated using a quartic polynomial

$$\begin{aligned} \text{s.t.} \quad & [s(0), \dot{s}(0), \ddot{s}(0)] = [s_0, \dot{s}_0, 0] \\ & [\dot{s}(T_F), \ddot{s}(T_F)] = [\dot{s}_i, 0]. \end{aligned}$$

For lateral movement, we sample the target offset $d(T_F) \leftarrow d_j$ in the range of $[-d_{lane}/2, d_{lane}/2]$, with d_{lane} denoting the lane width. Each lateral trajectory $d_j(t)$ is calculated using a quintic polynomial

$$\begin{aligned} \text{s.t.} \quad & [d(0), \dot{d}(0), \ddot{d}(0)] = [d_0, \dot{d}_0, 0] \\ & [d(T_F), \dot{d}(T_F), \ddot{d}(T_F)] = [d_j, 0, 0]. \end{aligned}$$

With the resulted longitudinal and lateral trajectory set $\mathcal{T}_{lon}$ and $\mathcal{T}_{lat}$, a full trajectory $\vec{x}(s(t), d(t))$ is formed by every combinations in $\mathcal{T}_{lon} \times \mathcal{T}_{lat}$. Here we impose the given constraints $\mathcal{C}$ to filter out infeasible trajectories. The first step is to project the Frenét coordinates (s, d) back to global coordinates (x, y), to check if the trajectory collides with static obstacles given in $\mathcal{C}_{\mathcal{M}}$. For any collision-free trajectory, its high-order state variables are then converted with the Frenét-Cartesian-transfomation

$$[s, \dot{s}, \ddot{s}, d, \dot{d}, \ddot{d}] \longmapsto [\vec{x}, v, \kappa, \alpha] \tag{2}$$

to check if its velocity v, acceleration α and curvature κ throughout the whole trajectory exceeds the kinematic constraints given in $\mathcal{C}_{tar}$. Here we refer interested readers to [40] for the detailed derivation of (2), and by this way, we ensure that each trajectory is kinematically feasible for the real-time state of a prediction target.

Finally, a bundle of trajectories $\{\mathcal{T}_{i,j} | j = 1, 2, ..., n_i\}$ compliant with environmental and kinematic constraints is generated from each reference path $\mathcal{P}_i \in \mathcal{P}$, and all the feasible trajectories together form an overall trajectory space $\mathcal{T}$. Although the constraints are set to be relatively conservative with leaving enough margin to the learning-based evaluator, our model-based generator effectively narrows down the trajectory space $\mathcal{T}$ by explicitly imposing constraints. This advantage would set our framework more stable in complex scenarios than the other data-driven frameworks.

V. Learning-based Evaluator

With the model-based generator G providing feasible trajectories in the first stage, the learning-based evaluator E is tasked to capture implicit agent-map interactions and select probable future trajectories as prediction results. In this section, we introduce a novel prediction evaluation network featured with a dual representation for spatial information. Our network aggregates information from scene context, including observed state sequences $\mathcal{S}$, a path set $\mathcal{P}$, and a future trajectory set $\mathcal{T}$. The overall framework is illustrated in Fig. 2.

A. State Representation

To make our network compatible with most existing trajectory prediction datasets, the state sequences $\mathbf{s}_i$ are reduced to history tracks in this section. Then before feeding to the network, each history track $\mathbf{s}_i$ and future trajectory $\mathcal{T}_i$ is discretized to a location sequence with time interval ΔT, and each reference path $\mathcal{P}_i$ is discretized as a waypoint sequence with distance interval ΔD. Since the longitudinal movement and lateral offset in (s, d) indicate how the agent moves relative to a reference path, it provides an understanding of the local spatial relationship in a more explicit way. For this reason, in addition to the commonly used Cartesian coordinates (x, y), we proposed to add the Frenét coordinates (s, d) to form a dual spatial representation for trajectory information. Here, the full information (x, y, s, d) of each future trajectory in $\mathcal{T}$ is given by the generator, while the (s, d) coordinates of each history track in $\mathcal{S}$ are obtained by projecting (x, y) coordinates on different reference paths. Additionally, we adopt the method from [22] to add a dimension of binary masks in the history track's representation (x, y, s, d, b) to indicate if the location at each timestamp is padded or not.

B. Encoding Scene Context

Prior to capture interrelationships between traffic entities, we first encode each kind of entity in the scene with a track encoder, a future encoder, and a path encoder. All encoders are structured with a temporal convolutional (1D Conv) layer followed by a long short-term memory (LSTM) layer. The track encoder and the future encoder employ a unidirectional LSTM and make the last hidden state $h(\cdot)$ as the motion encoding for history track and future trajectory, while the path encoder uses a bidirectional LSTM and provides the sequence of hidden states $H(\cdot)$ as the path spatial encoding. Given the scene context description $(\mathcal{S}, \mathcal{P}, \mathcal{T})$, each reachable path $\mathcal{P}_i \in \mathcal{P}$ is encoded as a $H(\mathcal{P}_i)$, where $i = 1, 2, ..., l$. Considering that the Frenét representation is dependent with the path frame, we encode all history tracks with respect to each reference path $\mathcal{P}_i$, which results in l groups of track encodings $\{h(\mathbf{s}_{tar}), h(\mathbf{s}_1), ..., h(\mathbf{s}_m)\}_i$. As the Frenét representation of each future trajectory $\mathcal{T}_{i,j} \in \mathcal{T}$ is relative to the corresponding path $\mathcal{P}_i$, so all future trajectories are encoded accordingly to form l groups of future encodings $\{h(\mathcal{T}_{i,j}) | j = 1, 2, ..., n_i\}$.

C. Modelling Interactions

Next, we propose several submodules to capture the implicit interactions existing in the scene context, which covers interrelationships between static environment and dynamic agents and interrelationships between multiple dynamic agents. We exploit the attention mechanism from [38] to fuse the spatial and temporal information from dynamic numbers of entities and then construct four modules using the most basic scaled dot-product attention, namely, path to tracks (P2T), path to

futures (P2F), agent to agent (A2A), and future to future (F2F). They are implemented in the same way, with linear layers for key mapping, query mapping, and value mapping. The overall workflow is shown in Fig. 2. In the upper branch, P2T brings the spatial information of each path encoding $\mathcal{P}_i$ into the corresponding track encodings $\{h(\mathbf{s}_{tar}), h(\mathbf{s}_1), ..., h(\mathbf{s}_m)\}_i$, and then the track encodings are further processed by a self-attention structure in A2A, aiming to capture the interactions between agents in the past time domain. The lower branch lays emphasis on updating the features contained in future encodings. P2F brings the spatial information of path encoding $H(\mathcal{P}_i)$ into the corresponding future encodings $\{h(\mathcal{T}_{i,j})|j = 1, 2, ..., n_i\}$. It is followed by F2F that fuses all future encodings $\bigcup_{i=1}^{l} \{h(\mathcal{T}_{i,j})|j = 1, 2, ..., n_i\}$ from different paths $\mathcal{P}_i (i = 1, 2, ..., l)$ using self-attention. In particular, F2F obtains a global understanding of the reachable space given by $\mathcal{P}$ and, by this way, attempts to further perceive the differences between different trajectories provided by $\mathcal{T}$. For any future trajectories $\mathcal{T}_{i,j} \in \mathcal{T}$, the corresponding track tensor $\mathbf{X}_i(\mathbf{s}_{tar})$, interaction tensor $\mathbf{Y}_i(\mathbf{s}_{tar})$ and future tensor $\mathbf{Z}(\mathcal{T}_{i,j})$ could be obtained from P2T, A2A and F2F modules, which are then concatenated together to form a full description $\mathbf{U}_{i,j} = \text{Concat}((\mathbf{X}_i(\mathbf{s}_{tar}), \mathbf{Y}_i(\mathbf{s}_{tar}), \mathbf{Z}(\mathcal{T}_{i,j}))$.

D. Trajectory Scoring and Learning

With obtaining $\mathbf{U}_{i,j}$ to describe the full information for a future $\mathcal{T}_{i,j}$, we score all the feasible trajectories $\mathcal{T}$ using a maximum entropy model:

$$\gamma(\mathcal{T}_{i,j}) = \frac{\exp(f(\mathbf{U}_{i,j}))}{\sum_{i=1}^{l} \sum_{j=1}^{n_i} \exp(f(\mathbf{U}_{i,j}))}, \tag{3}$$

in which $f(\cdot)$ is implemented using a 3-layer MLP at the end of the whole evaluation network E. The score label for each trajectory $\psi(\mathcal{T}_{i,j})$ is resulted from calculating the accumulated squared distance error Dist$(\cdot)$ between future trajectory $\mathcal{T}_{i,j}$ and ground truth trajectory $\mathcal{T}_{GT}$ in prediction horizon T_F:

$$\psi(\mathcal{T}_{i,j}) = \frac{\exp(-\text{Dist}(\mathbf{T}_{i,j}, \mathbf{T}_{GT})/\tau)}{\sum_{i=1}^{l} \sum_{j=1}^{n_i} \exp(-\text{Dist}(\mathbf{T}_{i,j}, \mathbf{T}_{GT})/\tau)}, \tag{4}$$

where τ is set as a temperature factor. Finally, the overall network is trained by cross entropy between the evaluated scores and the labeled scores $\mathcal{L} = \text{CrossEntropy}(\gamma(\mathcal{T}_{i,j}), \psi(\mathcal{T}_{i,j}))$.

E. Trajectory Selection

For the inference stage that requires a set of K trajectories as prediction results, we adopt the trajectory selection method from [41] to remove near-duplicate trajectories, which is derived from the non-maximum suppression (NMS) algorithm commonly used for object detection. According to predicted scores, this method greedily picks trajectories from $\mathcal{T}$ while excludes the lower scored trajectory between very close ones based on the distance metric Dist$(\cdot)$. Finally, K trajectories sorted in descending order of scores form the prediction results $\mathcal{T}_{tar} = \{\mathcal{T}_i|i = 1, 2, ..., K\}$, and the prediction probability p_i is derived by the corresponding scores, i.e., $p_i = \gamma(\mathcal{T}_i)/\sum_{i=1}^{K} \gamma(\mathcal{T}_i)$.

VI. Experiments

We evaluate our framework on the Argoverse motion forecasting benchmark [7], which provides a publicly available large-scale dataset for complex urban driving scenarios. In the following, we conduct extensive experiments, which covers: 1) comparing our approach against the state-of-the-art methods on the Argoverse test set; 2) ablation studies on some key components of our model; 3) verifying the feasibility of predicted trajectories; 4) evaluating the prediction robustness under imperfect observations. Lastly, some qualitative results and comparisons are demonstrated. We begin this section with an introduction to the experimental setup.

A. Experimental Setup

Dataset. Argoverse motion forecasting dataset [7] contains over 324K data sequence collected in traffic scenarios from Pittsburgh and Miami, which is split into 205942 sequences for training, 39472 sequences for validation, and 78143 sequences for testing. The training, validation, and test sets are taken from disjoint parts of the cities such that there is no geographical overlap. Each sequence lasts for 5 seconds, containing the centroid locations of each tracked object sampled at 10Hz, in which one tracked vehicle that follows relatively complex trajectories (such as changing lanes, navigating intersections, and turning) is marked as the prediction target. The objective is to predict a target vehicle's locations 3 seconds into the future, given an initial 2-second observation. The last 3 seconds of the test set are held out, so the model performance on that could only be assessed on the Argoverse official server. In addition to vehicle trajectories, the dataset provides an interface for accessing static map information, including lane centerlines and their connectivity.

Metrics. We follow the evaluation criteria of the Argoverse benchmark, which use the following metrics with prediction number $K = 1$ and $K = 6$:

- Miss Rate (MR_K): The ratio of scenarios that none of K predicted trajectories are within 2.0 meters of ground-truth according to endpoint error.
- Minimum Final Displacement Error (minFDE_K): The L2 distance between the endpoint of the *best* predicted trajectory and the ground-truth.
- Minimum Average Displacement Error (minADE_K): The average L2 distance between the *best* predicted trajectory and the ground-truth.

Here the endpoint denotes the predicted position at the last timestamp. Note that the Argoverse benchmark defines *best* the same in calculating minADE and minFDE, as the predicted trajectory with the minimum endpoint error. Additionally, we also add the probability-based metrics p-minFDE_K and p-minADE_K for multimodal predictions ($K = 6$). They are calculated by adding $(-log(p))$ with p-minFDE_K and p-minADE_K respectively, where p corresponds to the probability of the *best* predicted trajectory. The critical metric used for ranking the official leaderboard is MR_6, which reflects if a prediction model produces accurate multimodal predictions consistently in diverse challenging scenarios.

Method	K=1			K=6					Infeasibility (%)
	minADE	minFDE	MR (%)	minADE	minFDE	p-minADE	p-minFDE	MR (%)	
Argo-CV	3.53	7.89	83.48	3.39	7.57	5.18	9.36	81.68	0.00
Argo-LSTM+map	2.96	6.81	81.22	2.34	5.44	4.14	7.23	69.16	43.53
Argo-NN+map	3.65	8.12	83.55	2.08	4.03	3.87	5.82	58.21	86.39
LaneGCN [22]	1.71	**3.78**	59.05	**0.87**	**1.36**	2.66	3.16	16.34	16.52
Alibaba-ADLab	1.97	4.35	63.76	0.92	1.48	2.67	3.23	15.86	–
TNT [41]	1.78	3.91	59.72	0.94	1.54	2.73	3.33	13.28	–
Jean [25]	1.74	4.24	68.56	1.00	1.42	2.79	3.21	13.08	–
Poly (*3rd*)	**1.70**	3.82	58.80	0.87	1.47	2.67	3.26	12.02	–
S. Thomas-Huawei (*2nd*)	1.77	3.84	59.54	0.97	1.54	**2.56**	3.12	11.90	–
Ours-PRIME (***1st***)	1.91	3.82	**58.67**	1.22	1.56	2.71	**3.05**	**11.50**	**0.00**

TABLE I
COMPARISON WITH ARGOVERSE BASELINES AND TOP-RANKED ENTRIES ON THE ARGOVERSE MOTION FORECASTING LEADERBOARD. ALL METRICS ARE LOWER THE BETTER AND MISS RATE (MR, K=6) IS THE OFFICIAL RANKING METRIC.

Implementation Details. The implementation of our model-based generator and learning-based evaluator are detailed in the appendix (Sec. VIII). As for the state-of-the-art methods to be compared, only LaneGCN [22] is open-source. We use its official implementation and Argoverse baselines [7] for additional tests about trajectory feasibility and imperfect tracking.

B. Comparison with State of the Art

We compare our framework against the Argoverse official baselines [7] (CV, LSTM+map, NN+map), and top-ranked entries from the Argoverse leaderboard, including the recently published state-of-the-art methods: LaneGCN [22] from UberATG, and TNT [41] from Waymo & Google; the top-3 methods in Argoverse Motion Forecasting Competition 2020: Jean [25], Poly, and Alibaba-ADLab, in which Jean is the Winner of Argoverse 2020; as well as the latest top-3 methods by the data of 2021-03-01: Our PRIME, S. Thomas from Huawei, and Poly. The performance comparison under Argoverse test set is shown in Table I. It could be noted that the Argoverse Leaderboard is highly competitive with broad interest from both industrial and academy. For the core metric Miss Rate ($K = 6$) used for ranking, the previous leading entries are fairly close on this metric, which implies the difficulty of further improving it. Nonetheless, our PRIME outperforms all other methods by a large margin on MR_6 and achieves leading performance on the other metrics. Notably, our probability-based metric p-minFDE_6 is also the best among all methods, which would be highly beneficial to weigh between multiple predictions in making motion plans

Furthermore, from the methods with public details, including LaneGCN [22], TNT [41], and Jean [25], we can find that they all built upon a learning-based paradigm that utilizes neural networks to model all traffic entities and generate predictions. Our PRIME takes the 1st place with a novel framework design that integrates a model-based generator with a learning-based network. Notably, due to the lack of more detailed on-road information in the dataset, such as vehicle types, bounding box, static obstacles, etc., the current quantitative result is achieved by imposing relatively general constraints on our model-based generator, indicating there exists more space to improve when deploying our framework in a real autonomous driving system. Moreover, handling environmental and dynamical constraints in a model-based manner and generating continuous trajectories with full information is significant for real-world deployment, which could not be well reflected from the given metrics.

Modules	K=6			# Params
	p-minADE	p-minFDE	MR (%)	
Base	2.33	2.63	8.52	0.69 M
Base + F2F	2.31	2.61	8.23	0.72 M
Base + SD	2.29	2.58	7.81	0.99 M
Base + F2F + SD	**2.29**	**2.57**	**7.51**	1.02 M

TABLE II
ABLATION STUDIES ON THE ARGOVERSE VALIDATION SET.

C. Ablation Studies

We conduct ablation studies on the learning-based evaluator and report the probabilistic-based metrics (p-minADE, p-minFDE) and Miss Rate on the Argoverse validation set. With the P2T, P2A, and A2A attention modules capturing the basic interactions between map and agents, they form the base model together with the scene context encoders, while the F2F module and Frenét representation are respectively ablated from the full evaluation network to study their impacts.

From the results reported in Table II, we draw the following conclusions. First, the base model performs with $\text{MR}_6 = 8.52\%$, at the same level with TNT ($\text{MR}_6 = 9\%$ reported in [41]), reflecting the basic attention modules are effective in capturing the agent-map interactions from the scene context encodings. Second, as for the Frenét representation that provides the local relationship between path and trajectories and the F2F module that fuses all predictions to get a global

understanding of the reachable space, they both contribute to performance improvement. By comparison, the inclusion of Frenét representation is more effective for boosting the performance. Third, compared with other rasterization-based models [8, 26] with ResNet backbone (at least 11M parameters) and graph-based methods [14, 22], the overall model (Base + F2F + SD) makes the best performance with only 1.02M parameters, which indicates separating the function of trajectory generation would greatly simplify the network structure, and still, a high prediction performance could be achieved with our framework design.

D. Trajectory Feasibility

As the typical non-holonomic motion system, vehicles are constrained under inherent kinematic characteristics, while most prediction models generate unconstrained predictions by neural networks. So we investigate the ratio of infeasible trajectories of prediction models. Because the high-order information (velocity, acceleration, or turning rate) cannot be accurately estimated from discrete locations produced by common learning-based prediction models, we only evaluate the trajectory feasibility by curvature. By interpolating the predicted positions with pairwise cubic splines, we get the curvature at each point. Then a prediction is defined as infeasible if the curvature $\kappa > 1/3$ (i.e., the minimum turning radius is 3m) for any of its points. The ratio of infeasible predictions is shown in the last column of Table II. Except for the physical baseline Argo-CV (Constant Velocity), the others, as representatives of unconstrained models, have at least 16.5% predictions infeasible. Although we only use curvature for judgment and set the threshold very conservative,[2] the infeasible predictions still take up a considerable proportion, which would cause redundant burdens for SDVs to make decisions and plans. By contrast, the model-based generator in our framework can handle any kinematic and environmental constraints, thereby ensuring each trajectory prediction is feasible.

E. Imperfect Tracking

While most motion forecasting datasets provide tracking results of a certain duration for prediction targets, a self-driving vehicle would inevitably encounter some real-world situations where the target is lost in some timestamps or not tracked long enough yet. This requires the prediction model to robustly handle imperfect tracks rather than being restricted to fixed-duration tracking inputs. To let the models (ours, LaneGCN, and NN+map baseline) be aware of imperfect tracks, we retrained them by randomly dropping some tracked locations out. For processing such inputs while keeping original network structures, the locations of dropped timestamps are padded using the nearest tracked locations, and a dimension of binary masks is added to denote if the location is padded or not. Also, we retain the tracked location of the last timestamp to ensure the prediction target could be detected at inference time.

We randomly sample the drop rate from $0 \sim 0.6$ for each data sequence in training but fix it in testing. The drop rate is pointwise applied, i.e., a track with a 0.6 drop rate may drop more or less than 60% of locations. From the results shown in Fig. 4, we observe that our model performs stably, with only 4% relative increase on MR_6, while the relative increase is around $30\% \sim 40\%$ for the others. The result indicates that learning-based prediction models rely on long-term tracked results to regress trajectories, while our framework design relieves that to a certain extent, thereby improving the prediction robustness.

[2]The minimal turning radius for a regular sedan is around $4.5 \sim 6.0$m

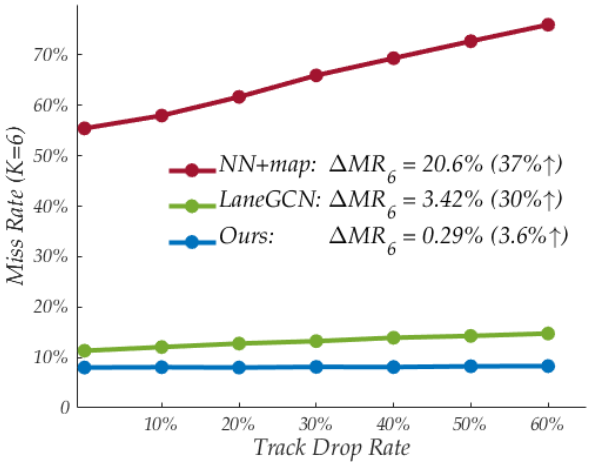

Fig. 4. Comparison of prediction robustness under imperfect tracking. We test how the MR_6 varies when the target's track is dropped at different rates.

F. Qualitative Results

We show our prediction results under diverse traffic scenarios in Fig. 5 and provide some representative comparisons with LaneGCN [22] in Fig. 6.

VII. Conclusion

We present a prediction framework PRIME that learns to predict vehicle trajectories with model-based planning. PRIME guarantees the trajectory feasibility by exploiting a model-based generator to produce future trajectories under explicit constraints. It makes accurate trajectory predictions by employing a learning-based evaluator to capture implicit interactions in scene context and select future trajectories by scoring. Furthermore, the combination of model-based trajectory generation and learning-based trajectory selection relieves high dependency on long-term tracking. Our PRIME outperforms other state-of-the-art prediction models in prediction accuracy, feasibility, robustness and achieves the 1st place on the highly-competitive Argoverse Leaderboard. More beyond these performance metrics, our approach reasonably regularizes the prediction space and produces trajectory predictions with continuous information, which facilitates decision making and motion planning for self-driving vehicles. Moreover, it is compatible with various on-road information, such as vehicle types, traffic rules, etc. These advantages would highly beneficial to deployments in real systems.

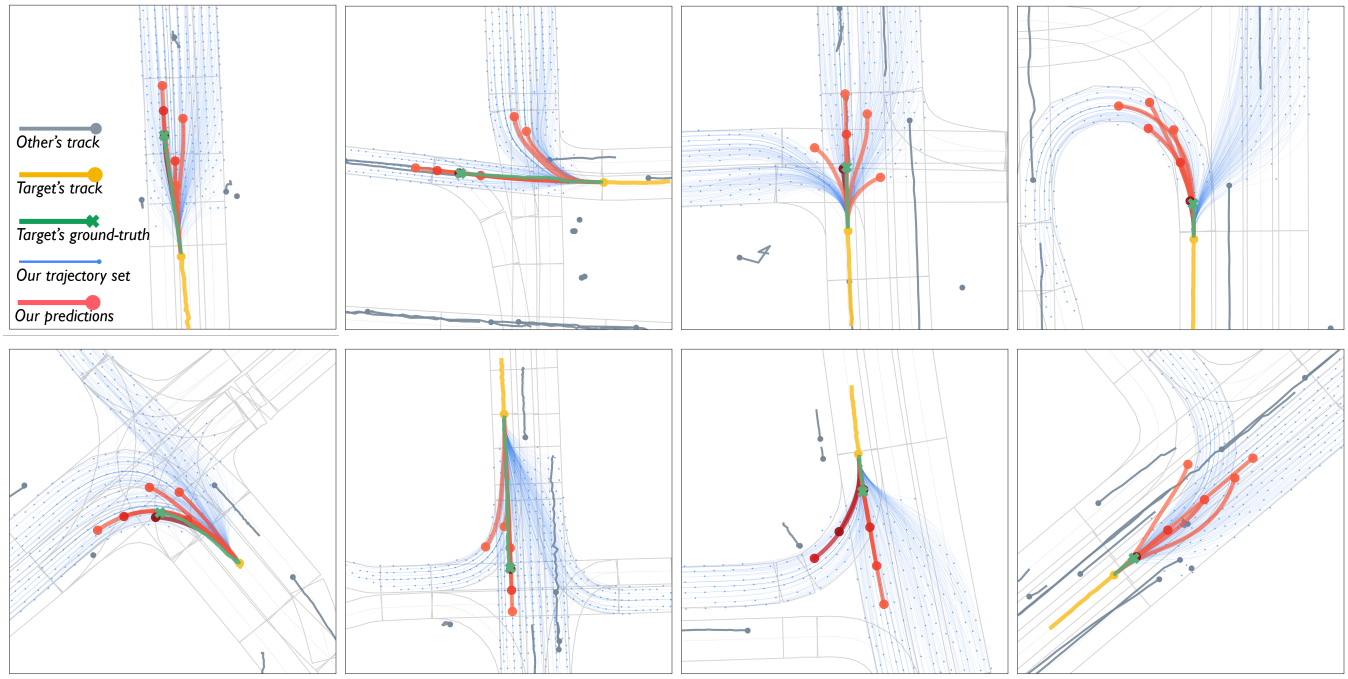

Fig. 5. Qualitative results under diverse scenarios on the Argoverse validation set. The HD map is depicted by light grey segments. The other agents' history tracks are shown in steel blue. The target agent's history track is shown in yellow and ground-truth future trajectory in green. The model-based generator produces the set of future trajectories $\mathcal{T}$ (blue) with feasibility guaranteed. The learning-based evaluator selects K trajectories from $\mathcal{T}$ as multimodal prediction results (red), and the depth of red indicates their probability.

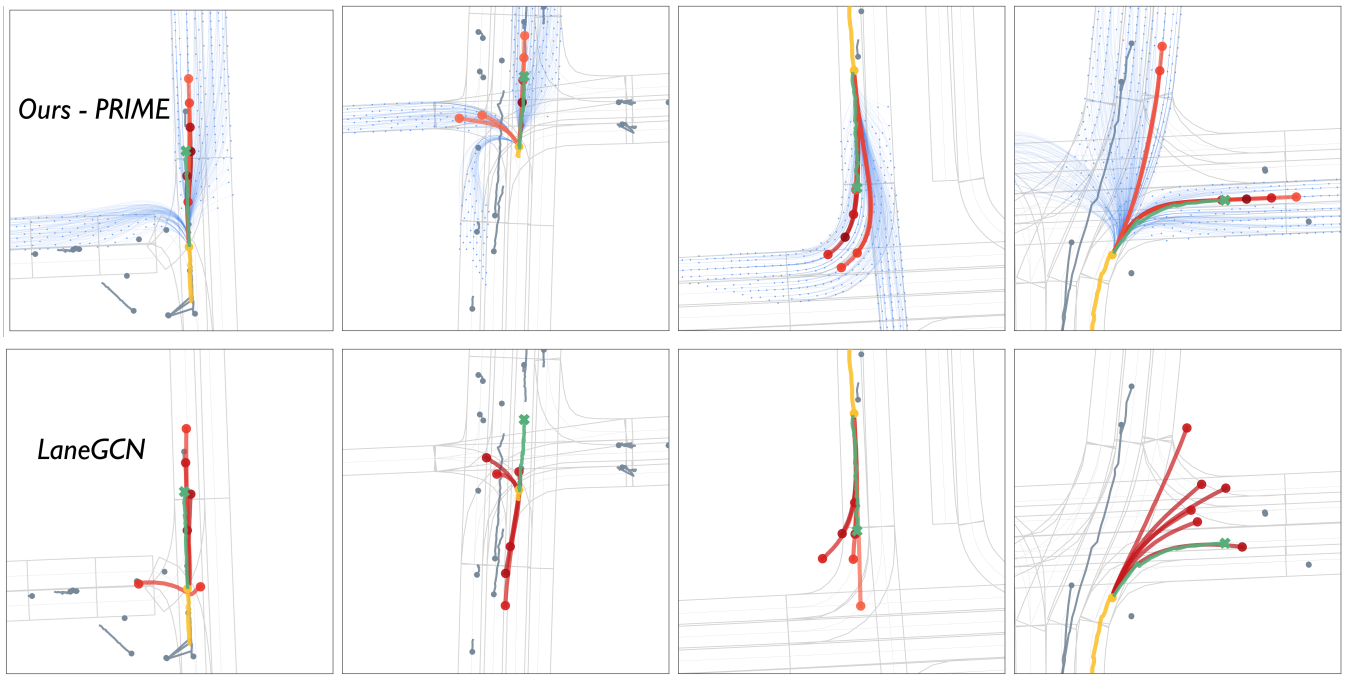

Fig. 6. Qualitative comparisons between ours (the upper row) and LaneGCN (the lower row) on the Argoverse validation set, with the same coloring scheme as in Fig. 5. Here, we use the state-of-the-art method, LaneGCN [22], as a representative for typical prediction models that generate unconstrained trajectories by neural networks. We show their common failures, including kinematically and environmentally infeasible predictions. Due to kinematic constraints, vehicles cannot take a turn suddenly at high speed (1st column), or reverse their moving directions (2nd column). Also, the prediction results of turning with across lane boundaries (3rd column), or heading towards reverse lanes (4th column) are incompliant with environmental constraints. Such infeasible predictions would cause redundant burdens for an AV to make decisions and motion plans. By contrast, the future trajectory set (blue) produced by our model-based generator is explicitly regularized by kinematic and environmental constraints, and thereupon, makes accurate and reasonable future predictions (red).

Rank	Participant team	minADE (K=1)	minFDE (K=1)	DAC (K=1)	MR (K=1)	minADE (K=6)	minFDE (K=6)	DAC (K=6)	MR (K=6)	p-minADE (K=6)	p-minFDE (K=6)	Last submission at
1	PRIME	1.9105	3.8217	0.9940	0.5867	1.2187	1.5582	0.9898	0.1150	2.7078	3.0473	9 days ago
2	Stefano_Thomas_Huawei_IOV_France (MRO_1)	1.7697	3.8380	0.9858	0.5954	0.9743	1.5425	0.9810	0.1190	2.5601	3.1284	3 days ago
3	poly	1.7003	3.8244	0.9908	0.5880	0.8735	1.4678	0.9854	0.1202	2.6653	3.2596	2 months ago
4	SenseTime_AP	1.6973	3.7573	0.9907	0.5830	0.8688	1.3639	0.9857	0.1203	2.6606	3.1557	3 months ago
5	LaneRCNN	1.6852	3.6916	0.9911	0.5685	0.9038	1.4526	0.9903	0.1232	2.6956	3.2443	3 months ago
6	TMP	1.7000	3.7758	0.9904	0.5835	0.8704	1.3688	0.9865	0.1300	2.6613	3.1596	4 months ago
7	Jean	1.7414	4.2372	0.9886	0.6856	0.9973	1.4209	0.9868	0.1308	2.7891	3.2127	8 months ago
8	tnt	1.7776	3.9143	0.9920	0.5972	0.9358	1.5384	0.9882	0.1328	2.7275	3.3301	7 months ago
9	TPCN	1.6372	3.6471	0.9855	0.5879	0.8587	1.3671	0.9836	0.1575	2.6166	3.1251	4 days ago
10	Alibaba-ADLab	1.9733	4.3478	0.9910	0.6376	0.9183	1.4831	0.9893	0.1586	2.6670	3.2318	8 months ago

Fig. 7. The top 10 entries on the Argoverse Motion Forecasting Leaderboard (snapshot made on 2021-03-01)

VIII. Appendix

A. Model-based Generator

1) State Estimation: : Argoverse provides vehicle state by discrete centroid positions, and there exists some noise in the data. We use Kalman Filter to estimate the target vehicle's velocity and heading and set its acceleration and turning rate to zero.

2) Path Search: : We use the Depth-First-Search algorithm to search potential paths that could be reached for prediction targets. The forward-searching distance D_F and backward-searching distance D_B are set to 140 and 20 meters separately. Each prediction target in the dataset is associated with 3.04 reachable paths on average.

3) Trajectory Generation: : For longitudinal movement, we sample the target velocity $\dot{s}(T_F)$ in the range of $[\max(0, \dot{s}_0 - \delta^- T_F), \min(\hat{\dot{s}}, \dot{s}_0 + \delta^+ T_F)]$ with $\hat{\dot{s}} = 30m/s$, $\delta^- = -6m/s^2$, $\delta^+ = 6m/s^2$ and the number of samples is set to 35. For lateral movement, we sample the target offset $d(T_F)$ in the range of $[-d_{lane}/2, d_{lane}/2]$, with the number of samples set to 9. Because the lane width cannot be queried from locations, we fix d_{lane} to 5 meters in lateral sampling.

With the generated longitudinal and lateral trajectory sets $\mathcal{T}_{lon}$ and $\mathcal{T}_{lat}$, a full trajectory $\vec{x}(s(t), d(t))$ is formed by every combinations in $\mathcal{T}_{lon} \times \mathcal{T}_{lat}$. Then we project the Frenét coordinates (s, d) back to global coordinates (x, y), to check trajectory's feasibility with respect to environmental constraints $\mathcal{C}_M$ and kinematic constraints $\mathcal{C}_{tar}$. Here, collision check is omitted as no static obstacle is given. The detailed vehicle information cannot be queried from the dataset either, so we adopt a general urban sedan setting for checking dynamic feasibility, with the maximum velocity $v = 33.33m/s$, maximum acceleration/deceleration $\alpha = \pm 8m/s^2$, and curvature $\kappa = 0.33$. If more detailed vehicle category information could be accessed, our future trajectory space could be further regularized. Consequently, each prediction target in the dataset obtains 484 feasible trajectories on average.

The runtime of generating a single trajectory by a single thread is around $0.1 \sim 0.2$ ms, and the generation process is highly parallelizable. So the model-based generator has a satisfactory real-time performance.

B. Learning-based Evaluator

For encoding scene context, all the continuous future trajectories $\mathcal{T}_i$ are discretized with time interval $\Delta T = 0.1$s, and all reachable paths $\mathcal{P}_i$ are discretized with distance interval $\Delta D = 2m$.

We train the evaluation network with a batch size of 64. The network is optimized using Adam with the learning rate initialized as 0.001 and decayed by 10 at every 10 epoch. We use Group Normalization with a group number of 4 for normalizing the data and LeakyReLU for non-linearity. We apply global random scaling with a scale ratio sampled from $0.75 \sim 1.25$ for data augmentation during training.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[2] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020.

[4] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Proc. Conference on Robot Learning (CoRL)*, pages 947–956. PMLR, 2018.

[5] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2020.

[6] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proc. Conference on Robot Learning (CoRL)*, 2019.

[7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8748–8757, 2019.

[8] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019.

[9] Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Jeff Schneider, David Bradley, and Nemanja Djuric. Deep kinematic models for kinematically feasible vehicle trajectory predictions. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 10563–10569. IEEE, 2020.

[10] Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1468–1476, 2018.

[11] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2095–2104, 2020.

[12] Dave Ferguson, Thomas M Howard, and Maxim Likhachev. Motion planning in urban environments. *Journal of Field Robotics*, 25(11-12):939–960, 2008.

[13] Enric Galceran, Alexander G Cunningham, Ryan M Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Proc. Robotics: Science and Systems (RSS)*, page 6, 2015.

[14] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11525–11533, 2020.

[15] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018.

[16] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[17] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8454–8462, 2019.

[18] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, and Wen Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pages 4363–4369. IEEE, 2013.

[19] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[20] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 336–345, 2017.

[21] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1–14, 2014.

[22] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph

representations for motion forecasting. In *Proc. European Conference on Computer Vision (ECCV)*, pages 541–556. Springer, 2020.

[23] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pages 759–776. Springer, 2020.

[24] Matthew McNaughton, Chris Urmson, John M Dolan, and Jin-Woo Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 4889–4895. IEEE, 2011.

[25] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 9638–9644. IEEE, 2020.

[26] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14074–14083, 2020.

[27] Mihail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

[28] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[29] Eike Rehder, Florian Wirth, Martin Lauer, and Christoph Stiller. Pedestrian prediction by planning using deep neural networks. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 5903–5908. IEEE, 2018.

[30] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proc. European Conference on Computer Vision (ECCV)*, pages 772–788, 2018.

[31] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 2821–2830, 2019.

[32] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1349–1358, 2019.

[33] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[34] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[35] Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[36] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. In *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[37] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pages 797–803. IEEE, 2010.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[39] Junqing Wei, John M Dolan, and Bakhtiar Litkouhi. A prediction-and cost function-based algorithm for robust autonomous freeway driving. In *2010 IEEE Intelligent Vehicles Symposium*, pages 512–517. IEEE, 2010.

[40] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 987–993. IEEE, 2010.

[41] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *Proc. Conference on Robot Learning (CoRL)*, 2020.

[42] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12126–12134, 2019.

[43] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pages 3931–3936. IEEE, 2009.

[44] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent transportation systems magazine*, 6(2):8–20, 2014.