2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Learning-Aided Warmstart of Model Predictive Control in Uncertain Fast-Changing Traffic

Mohamed-Khalil Bouzidi[1,2], Yue Yao[1,2], Daniel Goehring[1], Joerg Reichardt[2]

***Abstract*— Model Predictive Control lacks the ability to escape local minima in nonconvex problems. Furthermore, in fast-changing, uncertain environments, the conventional warmstart, using the optimal trajectory from the last timestep, often falls short of providing an adequately close initial guess for the current optimal trajectory. This can potentially result in convergence failures and safety issues. Therefore, this paper proposes a framework for learning-aided warmstarts of Model Predictive Control algorithms. Our method leverages a neural network based multimodal predictor to generate multiple trajectory proposals for the autonomous vehicle, which are further refined by a sampling-based technique. This combined approach enables us to identify multiple distinct local minima and provide an improved initial guess. We validate our approach with Monte Carlo simulations of traffic scenarios.**

I. Introduction

Model Predictive Control (MPC) has established itself as a popular technique in Motion Planning and Control for autonomous driving. This is attributed to its inherent capability to simultaneously account for collision constraints, dynamic feasibility, actuator constraints, and comfort criteria, enabling the generation of optimal trajectories [1]–[3]. A notable variant that we also use is Model Predictive Contouring Control (MPCC) [4]–[6]. It generates consistent lateral and longitudinal control signals and does not require a separate desired velocity specification. However, due to constrained computational resources, MPC relies on local optimization, employing simple models and limited planning horizons, potentially resulting in suboptimal or locally optimal (short-term) solutions. Conversely, learning-based approaches can excel where MPC falls short e.g., in efficiency and adaptability in complex tasks, without needing physical models [7], [8]. However, they face challenges in interpretability and reliability, especially in unexplored corner cases. This can potentially lead to hazardous behavior, hindering their suitability for critical applications. Hence, due to their complementary attributes, several methods propose approaches to combine MPC with learning-based approaches.

Learning-based MPC can be broadly categorized into two groups. The first group employs a learning-based system to substitute or enhance components of MPC. Simplest are approaches that learn the weights of the cost function [9], [10], as these significantly impact MPC performance and can be challenging to tune manually. A similar technique is cost shaping [11], [12] which adjusts the cost function at each time step, mitigating MPC's limitation in finding only short-term optimal solutions. Other methods learn the state-space model or parts of it [13]–[15] to handle unknown or complex dynamics. The second group learns high-level policies where the trajectory is further refined with low-level MPC. Methods such as [16], [17] provide high-level plans as a reference to the MPC. Similarly, the predictive safety filter [18], [19] evaluates constraint satisfaction of the trajectory of the learned system, potentially generating an output that minimizes the discrepancy from it while adhering to constraints.

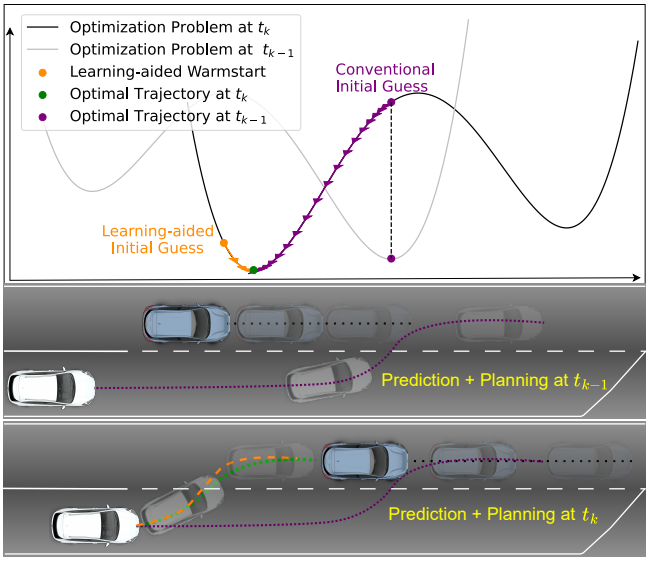

Fig. 1: Example where our warmstart improves convergence quality compared to warmstarting with the solution of the last timestep t_{k-1} due to change of the optimization problem (changing traffic participant behavior prediction)

Our approach of a learning-based warmstart also falls into this group, together with [20]–[23]. Here, the learned system offers an initial guess to the MPC optimizer, which is then further optimized by the MPC. This concept is particularly compelling given the inherent limitations of Local Optimizers/MPC, that become apparent in the context of autonomous driving in complex scenarios. The first well-known deficiency of the local optimizer is that if the initial guess is far from the optimum, many steps are needed until it converges, or the optimization may not converge at all. The strategy of MPC to provide an initial guess is to use the optimal trajectory, which was calculated in the last timestep, assuming little change between the previous and current

[1] is with the Free Universitity of Berlin, Germany
{firstname.lastname@fu-berlin.de}
[2] is with Continental AG
{firstname.lastname@continental.com}
This work received support by the German Federal Ministry for Economic Affairs and Climate Action within the project KI Wissen.

timestep. This strategy fails in uncertain and rapidly changing environments where the optimization problem can vary a lot between each timestep (s. Fig. 1). For instance, due to unknown intentions of human drivers, predictions of how traffic participants act may vary significantly between timesteps. These abrupt changes can lead the optimizer to struggle to recover or find a proper solution in time, potentially resulting in fatal behavior where e.g., collisions cannot be avoided. In the event of optimizer failure, a common approach is to use the same control input as in the last timestep. However, when the environment is changing rapidly, the scene can change even more in the time step after and now using the solution from two timesteps ago only exacerbates the problem.

The second deficiency of only being able to find a local optimum is especially problematic in dense traffic with (moving) obstacles. These obstacles are generally the cause for non-convex problems with multiple local minima. Some of these minima lead to undesired behavior, such as overly conservative driving or peculiar overtaking maneuvers. This problem is often mitigated by decomposing the planning problem into a global and a local planner [24], [25]. In this setup, the global planner generates a rough trajectory with significant simplifications for real-time feasibility, potentially sacrificing optimality. Also, topology-based planners such as [26]–[28] can address this weakness, from which we adopt the concept of homotopy classes. But, these planner also do not address the previously mentioned weakness of conventional warmstarting in fast-changing environments that our method tackles.

However, previous works for learning-based warmstarting [20]–[23] are mainly designed for simple repeating tasks. For example, they do not consider constraints, especially moving obstacles, or are trained for a limited number of self-generated scenarios (which additionally require retraining when the weights of the MPC cost function change).

The main contributions of our work are summarized as:

- Designing a Motion Planner based on Model Predictive Contouring Control with Artificial Potential Fields
- Developing a learning-aided warmstart strategy which improves convergence quality in fast-changing unknown scenarios and helps to prevent undesired local minima leveraging the concept of homotopy classes
- Devising a time-efficient framework with a novel trajectory refinement process which makes arbitrary multimodal trajectory predictors learned on real-world datasets easily deployable.

II. Baseline Model Predictive Contouring Control

Consider an arbitrary traffic scene with O traffic participants $o \in \{0, .., O-1\}$ in which an autonomous vehicle (AV), denoted as $o = 0$, needs to plan and execute a safe trajectory. A reference path and the map M_r with road boundaries are given by sets of waypoints $p_l = \{(x_j^l, y_j^l)\}_{j=0}^{K}$ with $l \in \{ref, lb, rb\}$ possibly provided by a high-level route planner. The reference path is parameterized by the arclength θ and augmented by the path

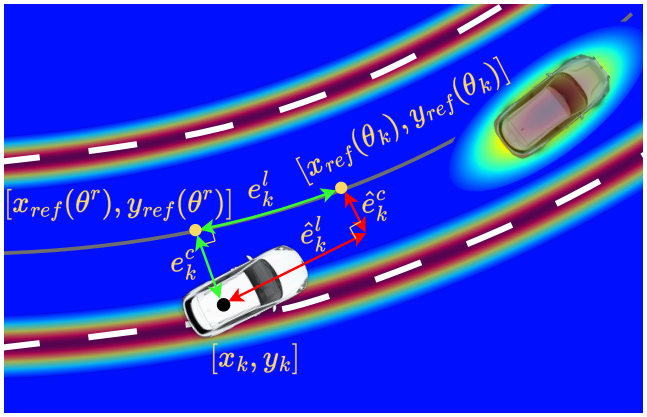

Fig. 2: Illustration of the lag error e_k^l, contouring error e_k^c (where θ^r is the real arclength) and potential field on obstacles and lane markers

orientation ψ_{ref} and the distance to the left and right road boundary d_{lb}, d_{rb}, $\mathcal{P}_{ref} : [0, \theta_{max}] \to \mathbb{R}^2 \times [0, 2\pi], \theta \mapsto (x_{ref}(\theta), y_{ref}(\theta), \psi_{ref}(\theta), d_{lb}(\theta), d_{rb}(\theta))$ where θ_{max} is the maximum arclength.

We model the motion of the AV by a differential equation $\dot{\boldsymbol{z}}(t) = f(\boldsymbol{z}(t), \boldsymbol{u}(t))$ using the kinematic bicycle model:

$$\dot{\boldsymbol{z}} = \left[v\cos(\psi),\ v\sin(\psi),\ v\frac{\tan(\psi)}{l},\ a,\ j,\ \dot{\delta} \right]^\top \tag{1}$$

where $\boldsymbol{z} = [x, y, \psi, v, a, \delta]^\top$ is the state vector, $\boldsymbol{u} = [j, \dot{\delta}]^\top$ is the control input vector and the velocity, acceleration, steering angle, jerk, steering angle rate, and wheelbase are denoted as $v, a, \delta, j, \dot{\delta}, l$, respectively.

For the MPC-formulation, the dynamic model is discretized to $\boldsymbol{z}_{k+1} = f(\boldsymbol{z}_k, \boldsymbol{u}_k)$ with the sampling time T_s. The MPCC aims to maximize path progress while minimizing path error, balancing between the two objectives. For that, we approximate the arclength (i.e. progress on the path) θ_k, the lag error $\hat{e}_k^l$ and the contouring error $\hat{e}_k^c$ (s. Fig 2):

$$\theta_{k+1} = \theta_k + v_k^p T_s \tag{2}$$

$$\begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \end{bmatrix} = \begin{bmatrix} \sin(\psi_{ref}(\theta_k)) & -\cos(\psi_{ref}(\theta_k)) \\ -\cos(\psi_{ref}(\theta_k)) & -\sin(\psi_{ref}(\theta_k)) \end{bmatrix} \Delta \boldsymbol{p}_{ref}$$

where $\Delta \boldsymbol{p}_{ref} = [x - x_{ref}(\theta_k),\ y - y_{ref}(\theta_k)]^\top$ and v_k^p is the virtual speed on the path. Eq. 2 is augmented to the dynamic model, i.e. v_k^p is an additional control input and θ_k a further state. This is utilized to define the running cost:

$$J_k = \begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \end{bmatrix}^\top Q \begin{bmatrix} \hat{e}_k^c \\ \hat{e}_k^l \end{bmatrix} - q_v v_k^p + \boldsymbol{u}_k^\top R \boldsymbol{u}_k \tag{3}$$

where Q, q_v, R are the respective weights. We extend the formulation of [4] to account for moving obstacles and lanes using the Potential Field method [2].

$$\begin{aligned} J_k^p = q_{ob} \sum_{i=0}^{O-1} \cdot \exp\left(-\left(\frac{\Delta x_k^o}{l^o}\right)^2 - \left(\frac{\Delta y_k^o}{w^o}\right)^2 \right) \\ + q_{lm} \sum_{l=0}^{L-1} \exp\left(-\left(\frac{d_{lm}^l - \hat{e}_k^c(\theta_k)}{\sigma}\right)^2 \right) \end{aligned} \tag{4}$$

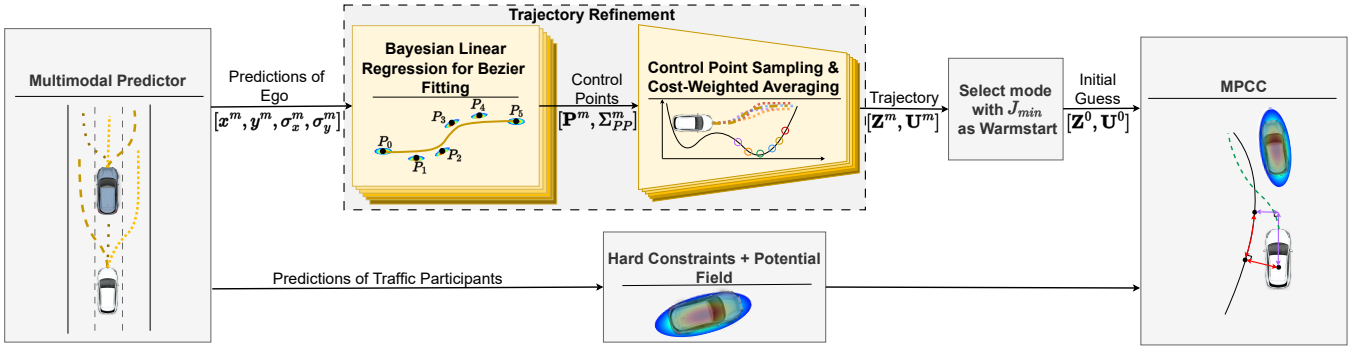

Fig. 3: The learning-aided warmstart framework for motion planning and control using a multimodal predictor

where $\Delta x_k^i, \Delta y_k^i$ are the distances to the respective obstacle, d_{lm}^l is the signed distance from the reference path to the respective L lane marker, σ a scaling factor and l^i, w^i a conservative estimation of the length and width of the obstacle and q_{ob}, q_{lm} are the respective weights. Additionally, for the hard constraints we employ ellipses to approximate the occupied area by the obstacles and utilize a union of three circles to approximate the ego's occupied space. With that, we approximate the Minkowsky sum as described in [5]. The trajectories of the obstacles are provided by the prediction module which will be introduced in the next section. To ensure that the AV stays within the road boundaries, we impose linear constraints

$$-d_{lb}\left(\theta_k\right) \leq \hat{e}_k^c\left(\theta_k\right) \leq d_{rb}\left(\theta_k\right). \tag{5}$$

Box constraints are imposed on the control inputs $j_k \in [j_{\min}, j_{\max}]$ and $\dot{\delta}_k \in [\dot{\delta}_{\min}, \dot{\delta}_{\max}]$. Additionally, we limit δ, a, and the lateral acceleration to ensure that the trajectories are feasible for the vehicle [29]. This leaves us with the nonconvex optimization problem:

$$\min_{\boldsymbol{Z},\boldsymbol{U}} \sum_{k=0}^{N-1} \left(J_k(\boldsymbol{z}_k, \boldsymbol{u}_k) + J_k^p(\boldsymbol{z}_k)\right) + J_N(\boldsymbol{z}_N) \tag{6}$$

$$\begin{aligned} s.t.\ \boldsymbol{z}_{k+1} &= f(\boldsymbol{z}_k, \boldsymbol{u}_k^H) \\ \boldsymbol{z}_0 &= \boldsymbol{z}(0) \\ \boldsymbol{z}_k \in \mathcal{Z},\ \boldsymbol{u}_k &\in \mathcal{U} \end{aligned}$$

where $\mathcal{Z}$ is set of state constraints imposed by road boundaries, obstacles, and lateral acceleration, $\mathcal{U}$ is the set of box constraints on the control inputs and $J_N(\boldsymbol{z}_N)$ is the terminal cost. The trajectories planned by the MPC are denoted by $\tau = [\boldsymbol{Z}, \boldsymbol{U}]^\top$ where $\boldsymbol{Z} = [\boldsymbol{z}_0, ..., \boldsymbol{z}_N]^\top$ and $\boldsymbol{U} = [\boldsymbol{u}_0, ..., \boldsymbol{u}_{N-1}]^\top$ where N is the prediction steps.

III. Learning-aided Warmstart

This section introduces our learning-aided approach, which we append as a warmstart method to the baseline MPCC introduced in sec. II (s. Fig. 3). The warmstart aims to provide an initial guess $\boldsymbol{\tau}^0 = [\boldsymbol{Z}^0, \boldsymbol{U}^0]$ sufficiently close to a satisfactory local optimum of the current time t_k, $||\boldsymbol{\tau}^0 - \boldsymbol{\tau}_k^*|| \leq \boldsymbol{\epsilon}$ such that the MPCC optimizer locally converges to this optimum. The conventional approach of warmstarting using the optimal trajectory of the last timestep $\boldsymbol{\tau}_{k-1}^* = [\boldsymbol{Z}_{k-1}^*, \boldsymbol{U}_{k-1}^*]$ is still employed in our framework by comparing it to the learning-aided output in terms of minimum cost at timestep k i.e., considering the new information e.g., about obstacle motion. This allows for enhancing the convergence quality while still maintaining an upper bound for the cost provided by $\boldsymbol{\tau}_{k-1}^*$.

A. Motion Predictor for Trajectory Proposals

Motion prediction models reason about the map, the historical trajectories of objects, and their interactions to forecast objects' future movement. However, determining the intentions of other traffic participants considering the various choices an agent can make (e.g. whether a car will overtake or follow a leading vehicle) is challenging. To address this challenge, many learning-based motion prediction models opt to provide *multimodal* predictions. Motion Transformer (MTR) [30] and Wayformer [31], are examples of such multimodal predictors trained on large-scale motion prediction datasets, such as Waymo Open Motion (WO) [32]. In our method, we employ MTR which outputs a Gaussian Mixture Model (GMM) for the object's future position $\mathcal{N}(\boldsymbol{\mu}^m, \boldsymbol{\Sigma}_{in}^m)$ at every timestep. Each component $m \in \mathcal{M}$ of this mixture corresponds to one predicted mode.

We capitalize on the necessity of the predictor for obstacle prediction[1] and reutilize it for predicting the ego trajectory. The aim of our approach is to leverage the multimodal output of the predictor to identify multiple local optima and select the best one. To elaborate on that, we introduce the concept of homotopy classes in the context of motion planning [33].

Definition 1. *Two trajectories connecting the same start and end position belong to the same homotopy class if they can be continuously deformed into each other without intersecting an obstacle. The set of all trajectories that are homotopic to each other is denoted as homotopy class.*

According to Definition 1, homotopic trajectories share the same start and end point. Due to the initial state constraint, the trajectories always share the same start by definition. However, we relax the end point requirement as suggested in

[1]In this study, we use the most probable obstacle prediction. Planning with multimodal obstacle predictions remains future work.

[34]. With the assumption that obstacles are the cause for the existence of multiple minima, it follows that all trajectories $\boldsymbol{\tau}_i \in \mathcal{A}$ are homotopic, where $\mathcal{A} = \{\boldsymbol{\tau} \in \mathcal{X} | J(\boldsymbol{\tau}^*) \leq J(\boldsymbol{\tau}_i), \boldsymbol{\tau}_i = \boldsymbol{\tau}^* + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \geq 0\}$ denotes the attractive vicinity of the local optimizer $\boldsymbol{\tau}^*$ [27].

Thus, the aim of the learning-aided warmstart can be further specified in providing an initial trajectory from the right homotopy class. One of the main causes of multimodality in motion prediction is the interaction with other traffic participants. Consequently, different modes often correspond to different homotopy classes. Therefore, we make the following assumptions:

Assumption 1. *Several of the predicted modes do not share the same homotopy class and cover a subset of the existing homotopy classes $h \in \mathcal{H}$, i.e. $|\{[m] | m \in \mathcal{M}\} \cap \mathcal{H}| \geq 2$.*

Assumption 2. *The covariance of the components of the GMMs i.e. for the respective mode is small enough such that trajectories drawn from the same components correspond to the same homotopy class (s. Fig. 4).*

Subsequently, we introduce how to utilize and further refine these provided modes to be able to select the best one (in terms of cost) as a warmstart.

B. Bezier Curve Fitting

Typical trajectory predictors such as MTR predict only the object position distributions at every prediction timestamp. However, we require the complete state and control input trajectories for our warmstart. Furthermore, our method is required to sample realistic trajectories from the given distribution to further refine the predictor trajectory.

We select 5th-degree Bezier Curves to fit the predicted positions. They represent the optimal solutions in terms of travel time, control effort, and jerk [35] and thus are close to the optimal vehicle trajectories outputted by the MPC. This smooths the often jerky predictions and allows us to calculate derivatives analytically. Further, we can perform this fit in such a way as to match current kinematic state. This continuity constraint is not directly enforced by MTR. We perform the fitting using Bayesian Linear Regression (BLR) to output a distribution over the curve parameters from which we can sample in the next step.

The 5th-degree Bezier Curve $\boldsymbol{c}(t)$ can be expressed as a linear combination of 6 control points $\boldsymbol{P}_j \in \mathbb{R}^2$ and the Bernstein polynomials $\phi_j(t) : \mathbb{R} \to \mathbb{R}$:

$$\boldsymbol{c}(t) = \sum_{j=0}^{5} \phi_j(t) \boldsymbol{P}_j \tag{7}$$

From the temporal derivatives of the Bezier curve, we can then calculate the state and control input trajectories.

Hence, we first need to estimate the control points $\boldsymbol{P}_j^m$ from the output of the predictor for each mode. The initial guess should ideally satisfy the continuity constraint $\boldsymbol{z}_0 = \boldsymbol{z}(0)$. For this, we exploit the property of the Bezier curve that the initial conditions can be determined from the control points. The initial condition for the (d^{th}) derivative can be calculated as:

$$\boldsymbol{c}^{(d)}(0) = \frac{6!}{(6-d)!} \Delta^d \boldsymbol{P}_0 \tag{8}$$

where Δ^k is the forward difference operator recursively defined by $\Delta^k \boldsymbol{P}_i = \Delta^{k-1} \boldsymbol{P}_{i+1} - \Delta^{k-1} \boldsymbol{P}_i$ where $\Delta^0 \boldsymbol{P}_i = \boldsymbol{P}_i$. From this, we determine the first three control points:

$$\begin{aligned} \boldsymbol{c}(0) =& [x_0, y_0]^\top, \quad \dot{\boldsymbol{c}}(0) = [v_0 \cos(\psi_0), v_0 \sin(\psi_0)]^\top, \\ \ddot{\boldsymbol{c}}(0) =& [a_0 \cos(\psi_0) - \frac{v_0^2}{l} \tan(\delta_0) \sin(\psi_0), \\ & a_0 \sin(\psi_0) - \frac{v_0^2}{l} \tan(\delta_0) \cos(\psi_0)]^\top \end{aligned} \tag{9}$$

We utilize these relationships for the control points as a strong Gaussian prior $\mathcal{N}(\boldsymbol{P}^{m,0}, \boldsymbol{\Sigma}^{m,0})$ for the BLR. In other words, the first three elements of $\boldsymbol{P}^{m,0}$ correspond to eq. 9, and $\boldsymbol{\Sigma}^{m,0}$ are derived from the tracked uncertainty of the states from the on-board sensors. As for the remaining three elements in $\boldsymbol{P}^{m,0}$, we employ an uninformed prior.

To formulate the BLR-problems we form $\mathbf{C}^m = \boldsymbol{\Phi}^\mathrm{T} \boldsymbol{P}^m$ with the vector of the control points $\boldsymbol{P}^m \in \mathbb{R}^{6 \cdot 2}$, the vector of Bezier curve points $\boldsymbol{C}^m \in \mathbb{R}^{2N}$ and the new basis function $\boldsymbol{\Phi} \in \mathbb{R}^{6 \cdot 2 \times 2N}$ as done in [36], [37]. The uncertainties of the predictions $[\boldsymbol{X}_m, \boldsymbol{Y}_m]^\top$, i.e. the covariance $\boldsymbol{\Sigma}_{in}^m$ outputted by the GMM enter as Gaussian observation noise into the regression.

$$[\boldsymbol{X}_m, \boldsymbol{Y}_m]^\top = \boldsymbol{\Phi}^\top \boldsymbol{P}^m + \boldsymbol{e}, \ \boldsymbol{e} \sim N(0, \boldsymbol{\Sigma}_{in}^m) \tag{10}$$

Consequently, the posterior and the covariance for the control points are given:

$$\boldsymbol{\Sigma}_{PP}^m = \left(\boldsymbol{\Phi} \boldsymbol{\Sigma}_{in}^m \boldsymbol{\Phi}^\top + \left(\boldsymbol{\Sigma}^{m,0} \right)^{-1} \right)^{-1} \tag{11}$$

$$\boldsymbol{P^m} = \boldsymbol{\Sigma}_{PP}^m \boldsymbol{\Phi}^\top \left(\boldsymbol{\Sigma}_{in}^m \right)^{-1} \begin{bmatrix} \boldsymbol{X}_m \\ \boldsymbol{Y}_m \end{bmatrix} + \boldsymbol{\Sigma}_{PP}^m \left(\boldsymbol{\Sigma}^{m,0} \right)^{-1} \boldsymbol{P}^{m,0}$$

C. Control Point Sampling and Cost-Weighted Averaging

Simply calculating the states and control inputs from each outputted modes of the predictor leads often to an ineffective warmstart. Even if the best homotopy class is chosen from these modes, it can still lead to a solution far from the optimum, resulting in a prolonged convergence time. Additionally, comparing modes to select the best homotopy class based on the cost function is inaccurate, as for two trajectories in two different homotopy classes $\boldsymbol{\tau}_{h,1}, \boldsymbol{\tau}_{h,2}$ $J(\boldsymbol{\tau}_{h,1}) > J(\boldsymbol{\tau}_{h,2})$ does not necessarily imply $J(\boldsymbol{\tau}_{h,1}^*) > J(\boldsymbol{\tau}_{h,2}^*)$ since $\boldsymbol{\tau}_{h,1}$ and $\boldsymbol{\tau}_{h,2}$ can have different distances from their local optimum $\boldsymbol{\tau}_{h,1}^*$ and $\boldsymbol{\tau}_{h,2}^*$. Our solution to refine the trajectories is to utilize the distributions $\mathcal{N}(\boldsymbol{P}^m, \boldsymbol{\Sigma}_{pp}^m)$ from eqn. 11, in order to sample S Bezier curves (s. Fig. 4). For each sample $\boldsymbol{P}_s^m$ and the mean $\boldsymbol{P}^m$, we compute the cost function and obtain our output control points through a cost-weighted average.

$$\bar{\boldsymbol{P}}^m = \sum_{s=0}^{S} w_s \tilde{\boldsymbol{P}}_s^m \text{ with } w_s = \frac{\mathrm{e}^{-\lambda J(\tilde{\boldsymbol{P}}_s^m)}}{\sum_{i=0}^{S} \mathrm{e}^{-\lambda J(\tilde{\boldsymbol{P}}_i^m)}} \tag{12}$$

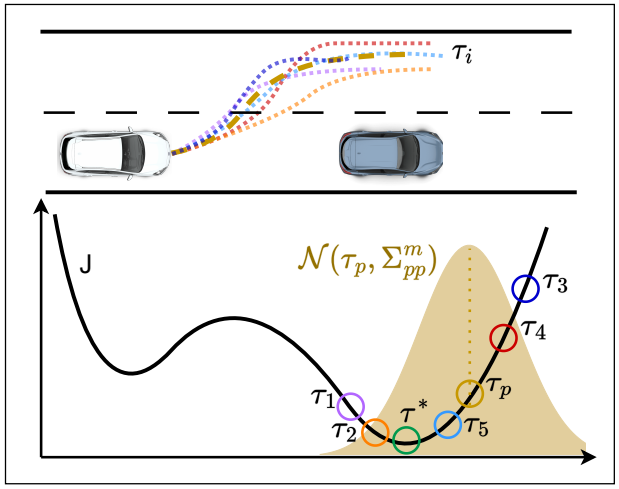

Fig. 4: Intuition of the trajectory sampling around the fitted trajectory $\boldsymbol{\tau}_p$ of the predictor which is assumed to fall in the attractive vicinity of local minimum

where λ is a tunable parameter. The chosen weighting factor w_n has the advantageous property that assigns significantly less influence to samples with markedly higher cost function compared to the sample with the minimum cost, i.e., $J(\tilde{\boldsymbol{P}}_i^m) >> J(\tilde{\boldsymbol{P}}_j^m), w_i \to 0$ (softmin normalization [38]). This implies that, in essence, we are only giving substantial consideration to a limited subset of samples within a similar cost range.

Furthermore, recall Assumption 2, i.e.; consequently, we assume the samples are in the attractive vicinity of a local minimum. Provided the samples are well distributed around the region of convexity of this local optimum, taking the weighted average of the trajectories gives us a value inside the area spanned by the sample points. Hence, the output results generally in a trajectory closer to the minimum.

We execute this process for each mode. Subsequently, the costs for each mode are compared, and the best one is employed as the warmstart. While this still does not guarantee the selection of the homotopy class of the global optimum, it allows us to choose a satisfactory local minimum at least. In autonomous driving, various maneuvers are often similarly satisfactory, and only undesired local minima must be prevented.

The detailed steps of the trajectory refinement are outlined in Algorithm 1. It is important to note that this approach supports parallel computation due to the parallel nature of sampling and the independence of each trajectory from one another, i.e., it can take advantage of the parallel processing capabilities of modern GPUs, making it highly efficient.

Algorithm 1: Learning-aided Warmstart

Input : Measured state values
$\boldsymbol{z}_k = [x_k, y_k, \psi_k, v_k, a_k, \delta_k, \theta_k]^\top$,
optimal traj.last timestep $\boldsymbol{Z}_{k-1}^*$, $\boldsymbol{U}_{k-1}^*$,
map information $\boldsymbol{M_r}$, reference path $\mathcal{P}_{ref}$,
pose history $\boldsymbol{\eta}_o^k \; \forall$ Agents o with $o = 0$
denoting the ego vehicle
Output: Initial Guess for MPCC $\boldsymbol{Z}^0$, $\boldsymbol{U}^0$
Initialize # refinement samples S, # used modes M
// Predict trajectories $\boldsymbol{\xi}_o^m = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y]$
$\boldsymbol{\xi} \leftarrow \mathcal{MTR}(\boldsymbol{\eta}^k, \boldsymbol{M_r})$
foreach $m = \{1, ..., M\}$ **do**
  // Fit predicted ego trajs. into Bezier Curve
  $\boldsymbol{P}^m, \boldsymbol{\Sigma}_{pp}^m \leftarrow \text{BayesReg}(\boldsymbol{\xi}_0^m, \boldsymbol{z}_k)$ from (11)
  // Sample from $\mathcal{N}(\boldsymbol{P}^m, \boldsymbol{\Sigma}_{pp}^m)$
  $\tilde{\boldsymbol{P}}_1^m, \ldots, \tilde{\boldsymbol{P}}_S^m \sim \mathcal{N}(\boldsymbol{P}^m, \boldsymbol{\Sigma}_{pp}^m)$
  foreach $s = \{1, ..., S\}$ **do**
    // Calculate States, Control Inputs and Cost
    $\boldsymbol{Z}_s^m, \boldsymbol{U}_s^m, J_s^m \leftarrow \mathcal{J}(\tilde{\boldsymbol{P}}_s^m, \mathcal{P}_{ref}, \boldsymbol{\xi}_{1:O}^0)$
  // Calculate Cost-weighted Average of samples
  $\bar{\boldsymbol{P}}^m \leftarrow$ from (12)
  // Calculate States, Control Inputs and Cost
  $\boldsymbol{Z}^m, \boldsymbol{U}^m, J^m \leftarrow \mathcal{J}(\bar{\boldsymbol{P}}^m, \mathcal{P}_{ref}, \boldsymbol{\xi}_{1:O}^0)$
// Select mode with minimal cost
$m^* \leftarrow \arg\min_m J^m$
if $J^{m^*} \leq$ *Cost of* $\boldsymbol{Z}_{k-1}^*, \boldsymbol{U}_{k-1}^*$ *at timestep* k **then**
  return $(\boldsymbol{Z}^0 \leftarrow \boldsymbol{Z}^{m^*}, \boldsymbol{U}^0 \leftarrow \boldsymbol{U}^{m^*})$
else
  return $(\boldsymbol{Z}^0 \leftarrow \boldsymbol{Z}_{k-1}^*, \boldsymbol{U}^0 \leftarrow \boldsymbol{U}_{k-1}^*)$

IV. Performance Evaluation

We compare our MPCC with learning-aided warmstart to the baseline MPCC with conventional warmstart in three experiments. The first two experiments serve as illustrative examples to highlight the two strengths of our approach. The last experiment entails a Monte Carlo simulation of random highway merging scenarios to provide statistical results (s. Tab. I).

Experiment I involves a scenario with two lanes, where the left lane accommodates oncoming traffic but allows for overtaking. This scenario is well-suited to showcase the capability of our framework in escaping undesired local minima, as the presence of other traffic participants introduces non-convexity to the optimization problem. In [28], [34], it is demonstrated that a planner in this scenario may converge towards several distinct local minima/homotopy classes. In our case, our learning-aided warmstart leads to a different behavior compared to the MPCC without warmstart (s. Fig. 5); i.e., the two planners converge towards two distinct local optima. The costs for our planner are significantly lower than those for the baseline planner (s. Fig. 5). This figure also provides a comparison of the control input trajectories and the minimum time-to-collision (TTC) for both planners in this scene, displaying the shortcomings of the local minima the baseline converged to.

Experiment II involves a scenario where an obstacle crosses the path of the ego vehicle. Initially, the ego vehicle is unaware of this occurrence for the first few moments, causing a sudden shift in the optimization problem for the planner. Such a situation can arise in various scenarios, for instance, when the obstacle is initially occluded or when predictions change (due to unknown intentions of traffic participants or

TABLE I: Results of experiment III. Comparison of Baseline and the learning-aided Framework using Monte Carlo analysis

	Merging Execution			Convergence Quality				
	Success	Aborted	Collision	Success	Max. Time exceeded	Converge to Infeasibility	Average Cost	Average Solving time (std)
Baseline MPCC	73 %	11%	16%	69.3%	13.6%	17.1%	4995	106 ms (40 ms)
Our Framework	**88 %**	**4%**	**8%**	**82.7%**	**7.0%**	**10.3%**	**3737**	**94 ms (33 ms)**

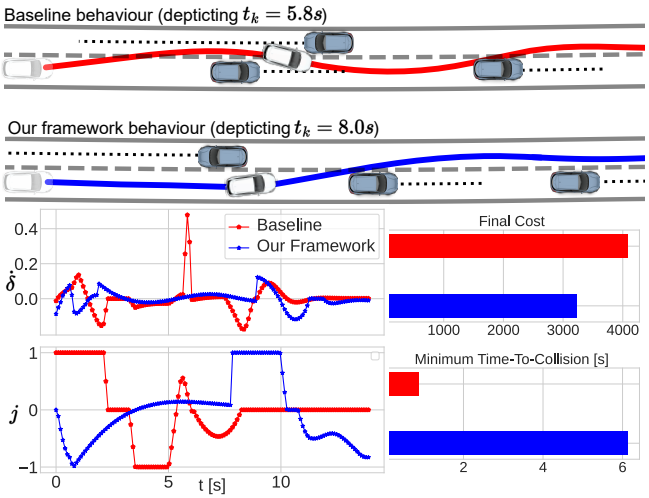

Fig. 5: Illustration of experiment I. Comparison of the local minimum that the baseline converged to with our framework

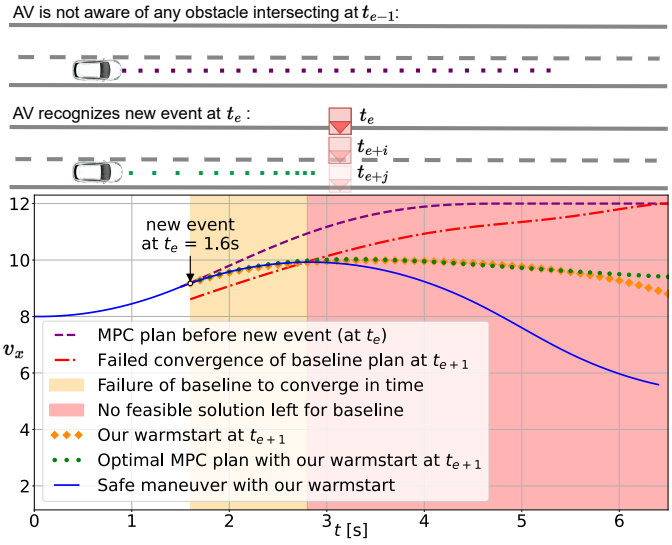

Fig. 6: Illustration of experiment II. Comparison of the baseline to our framework when an event occurs that changes the optimization problem between two timesteps

a new decision of an object). In this case, the planner must be capable of finding a solution for the new optimization problem in real-time, even though it differs distinctly from the last timestep. Hence, we impose a maximum solving time constraint. However, we set this limit relatively high with $t_{\max} = 0.5s$ since there is potential to accelerate the MPC runtime through alternative implementations and hardware enhancements, etc. Despite this high maximum solving time, the baseline planner is unable to converge in time when the change occurs (from the new event at $t_e = 1.6$ in Fig. 6). The red curve depicts the velocity trajectory outputted by the solver. This trajectory fails to satisfy both collision constraints and the initial condition. In such cases, it is customary to utilize the solution from the last time step, which, in this example, leads to further acceleration of the ego. This behavior ultimately results in an unavoidable collision. In contrast, Fig. 6 depicts that our approach can directly provide an appropriate warmstart after the event.

For experiment III, we consider a highway merging scenario (s. Fig. 1) and utilize the Intelligent Driver Model [39] to simulate the behavior of the traffic participants. We generate 100 test runs by randomly sampling the parameters of the IDM model (such as desired velocity, minimum headway, etc.), as well as the initial positions and velocities for the ego vehicle and the other vehicles. As a result, we compare the rate of successful mergings, the percentage of the ego getting stuck in the entrance lane, and collisions. Additionally, we assess the convergence quality in terms of the percentage of successful convergence, failed convergence due to reaching the time limit, and failed convergence due to converging to a point of infeasibility. Further benchmarking parameters are the average cost and solving time[2]. The significant performance improvement to the baseline becomes evident when considering highway merging. Firstly, each gap between traffic participants potentially corresponds to a local minimum where one can clearly be better than the other e.g., due to gap size. Secondly, shifts in motion predictions of the traffic participants during merging often substantially impact the ego vehicle's plan e.g., if a prediction changes the acceleration slightly, the optimal plan for the ego may shift from merging in front to merging behind.

V. CONCLUSIONS

A Learning-aided Warmstart Framework is proposed to address the problem of Model Predictive Control with local minima and convergence issues if using the conventional warmstart strategy in fast-changing, uncertain environments. This framework leverages a multimodal predictor that predicts trajectories for traffic participants and the ego vehicle, respectively. The different ego trajectory modes are used to identify multiple homotopy classes, each associated with an attractive vicinity of a different local optimum. To achieve this, we introduced a novel sampling-based trajectory refinement approach using Bayesian Linear Regression for Bezier Curve Fitting to efficiently optimize the trajectories before selecting the best one as an initial guess. Our Monte Carlo analysis demonstrates that our framework significantly improves the convergence quality in highway merging scenarios.

[2]Solving time data is for comparative purposes only and should not be taken as absolute

REFERENCES

[1] F. Micheli, M. Bersani, S. Arrigoni, F. Braghin, and F. Cheli, "NMPC trajectory planner for urban autonomous driving," *Vehicle system dynamics*, vol. 61, no. 5, pp. 1387–1409, 2023.

[2] F. Siebenrock, M. Günther, and S. Hohmann, "LTV-MPC Based Trajectory Planning Considering Uncertain Object Prediction Through Adaptive Potential Fields," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2020, pp. 666–672.

[3] Q. Shi, J. Zhao, A. El Kamel, and I. Lopez-Juarez, "MPC Based Vehicular Trajectory Planning in Structured Environment," *IEEE Access*, vol. 9, pp. 21 998–22 013, 2021.

[4] A. Liniger, A. Domahidi, and M. Morari, "Optimization-Based Autonomous Racing of 1:43 Scale RC Cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[5] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[6] L. Lyons and L. Ferranti, "Curvature-Aware Model Predictive Contouring Control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3204–3210.

[7] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2022.

[8] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," 2018.

[9] S. Gros and M. Zanon, "Data-driven Economic NMPC using Reinforcement Learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2019.

[10] B. Zarrouki, V. Klös, N. Heppner, S. Schwan, R. Ritschel, and R. Voßwinkel, "Weights-varying MPC for Autonomous Vehicle Guidance: a Deep Reinforcement Learning Approach," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 119–125.

[11] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go Next: Learning a Subgoal Recommendation Policy for Navigation in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4616–4623, 2021.

[12] A. Tamar, G. Thomas, T. Zhang, S. Levine, and P. Abbeel, "Learning from the Hindsight Plan – Episodic MPC Improvement," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 336–343.

[13] T. Koller, F. Berkenkamp, M. Turchetta, J. Bödecker, and A. Krause, "Learning-based model predictive control for safe reinforcement learning," in *Robust autonomy: tools for safety in real-world uncertain environments*, 2019.

[14] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-Driven Model Predictive Control With Stability and Robustness Guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[15] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[16] B. Brito, A. Agarwal, and J. Alonso-Mora, "Learning Interaction-aware Guidance Policies for Motion Planning in Dense Traffic Scenarios," *arXiv preprint arXiv:2107.04538*, 2021.

[17] Y. Song and D. Scaramuzza, "Policy Search for Model Predictive Control with Application to Agile Drone Flight," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2114–2130, 2022.

[18] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[19] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7635–7642, 2021.

[20] N. Mansard, A. DelPrete, M. Geisert, S. Tonneau, and O. Stasse, "Using a Memory of Motion to Efficiently Warm-Start a Nonlinear Predictive Controller," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2986–2993.

[21] T. Barbié, R. Kabutan, R. Tanaka, and T. Nishida, "Gaussian mixture spline trajectory: learning from a dataset, generating trajectories without one," *Advanced Robotics*, vol. 32, no. 10, pp. 547–558, 2018.

[22] S. Natarajan, "Learning initial trajectory using sequence-to-sequence approach to warm start an optimization-based motion planner," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9430–9436.

[23] T. S. Lembono, A. Paolillo, E. Pignat, and S. Calinon, "Memory of Motion for Warm-starting Trajectory Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2594–2601, 2020.

[24] M. Svenstrup, T. Bak, and H. J. Andersen, "Trajectory planning for robots in dynamic human environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4293–4298.

[25] J. Li, M. Ran, H. Wang, and L. Xie, "MPC-based Unified Trajectory Planning and Tracking Control Approach for Automated Guided Vehicles," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 374–380.

[26] O. de Groot, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Globally Guided Trajectory Planning in Dynamic Environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 118–10 124.

[27] C. Rösmann, F. Hoffmann, and T. Bertram, "Planning of multiple robot trajectories in distinctive topologies," in *2015 European Conference on Mobile Robots (ECMR)*. IEEE, 2015, pp. 1–6.

[28] B. Yi, P. Bender, F. Bonarens, and C. Stiller, "Model Predictive Trajectory Planning for Automated Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 24–38, 2018.

[29] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "Guaranteeing Consistency in a Motion Planning and Control Architecture Using a Kinematic Bicycle Model," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 3981–3987.

[30] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

[31] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion Forecasting via Simple & Efficient Attention Networks," *arXiv preprint arXiv:2207.05844*, 2022.

[32] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[33] S. Bhattacharya, *Topological and geometric techniques in graph search-based robot planning*. University of Pennsylvania, 2012.

[34] P. Bender, Ö. Ş. Taş, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1386–1392.

[35] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.

[36] Y. Yao, D. Goehring, and J. Reichardt, "An empirical bayes analysis of vehicle trajectory models," *arXiv preprint arXiv:2211.01696*, 2022.

[37] J. Reichardt, "Trajectories as markov-states for long term traffic scene prediction," in *14-th UniDAS FAS-Workshop*, Berkheim, Germany, 2022, p. 14.

[38] M. D. Houghton, A. B. Oshin, M. J. Acheson, E. A. Theodorou, and I. M. Gregory, "Path planning: Differential dynamic programming and model predictive path integral control on vtol aircraft," in *AIAA SCITECH 2022 Forum*, 2022, p. 0624.

[39] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.