A Fully Differentiable Framework for Autonomous Driving Based on Model-Structured Neural Networks

Sabrina Ciuffoletti, Gioele Defrancesco, Giovanni Scialla, Mattia Piazza, Sebastiano Taddei, Gastone Pietro Rosati Papini

Abstract— **Generalization in autonomous driving remains an open challenge, with current approaches divided between modular pipelines, which ensure interpretability and safety, and end-to-end learning methods, which offer flexibility but limited guarantees. This work proposes a hybrid framework that combines these paradigms through Model Structure Neural Networks (MSNNs) within a fully differentiable architecture. The system integrates vehicle dynamics modeling, control, and motion planning into a unified pipeline. A structured neural forward model is first identified from data, followed by the learning of an inverse controller through a closed-loop, mental simulation approach. A neural planner, based on a modular motion planning structure, is then connected to the control layer, enabling end-to-end training and joint optimization. The entire process is supported by nnodely, an open-source framework for the design and training of MSNNs. The paper presents the structure of each module and the integration within the control pipeline. The complete system is validated in closed-loop simulations in CarMaker, where the planner network is directly connected to the inverse model to control the vehicle. Results highlight the potential of structured neural approaches for achieving robust and generalizable autonomous driving.**

I. Introduction

Achieving robust task generalization remains one of the central open challenges in autonomous driving. While recent advances in machine learning have enabled impressive perception and control capabilities, current systems still struggle when deployed in unseen environments or under distribution shifts. This limitation has led to the emergence of two dominant and often contrasting paradigms: on one side, modular architectures grounded in engineering principles and designed for interpretability and safety [1]. In contrast, end-to-end learning approaches that leverage large-scale data to learn complex behaviors directly from experience [2]. Despite their respective successes, neither paradigm alone has yet demonstrated the ability to reliably generalize across the wide variability of real-world driving conditions [3]. In this context, hybrid approaches that combine structure and learning are gaining increasing attention. Model Structure Neural Networks offer a promising direction by embedding prior knowledge directly into neural architectures(*i.e.,* physical laws and control principles). This integration introduces inductive biases that improve data efficiency, interpretability, and robustness, while retaining the adaptability of learning-based methods [4], [5]. At the same time, recent work on learning control through mental simulation has shown that it is possible to construct predictive and control models in a fully data-driven yet structured manner [6], by first learning forward dynamics and then synthesizing inverse models through differentiable simulation pipelines. These ideas suggest a pathway toward autonomous systems that are both learnable and controllable, bridging the gap between model-based and data-driven paradigms. This paper builds on these insights and proposes a unified framework for autonomous driving based on fully differentiable, model-structured neural architectures. The approach integrates three key components, *e.g.,* vehicle dynamics modeling, control, and motion planning, within a single end-to-end trainable pipeline. First, a neural forward model of the vehicle dynamics is identified from data using a structured architecture that reflects the underlying physics. Then, an inverse neural controller is synthesized by coupling it with the forward model and training the two in a closed loop, following the mental simulation paradigm. Finally, a high-level planner, inspired by modular motion planning approaches [6], is introduced as a neural component that generates target trajectories and is trained jointly with the control layer through the differentiable vehicle model. Moreover, this hierarchical yet fully differentiable structure enables joint optimization and fine-tuning across all levels of the autonomy stack. Once trained, the control networks are exported for deployment in real-world applications. Importantly, their neural nature allows, in principle, for further online adaptation. A key enabler of this framework is nnodely, an open-source environment specifically designed for the development of MSNNs. By providing modular building blocks aligned with physical modeling and control concepts, nnodely allows the construction of structured neural architectures that can be trained, interconnected, and deployed within a unified workflow. Importantly, the framework supports the composition of multiple neural modules into larger differentiable systems, enabling end-to-end learning across perception, planning, and control layers. The proposed approach directly addresses the core question of this workshop: how to achieve generalization in autonomous driving. By combining structured inductive biases with data-driven learning, it aims to overcome the limitations of both purely modular and purely end-to-end systems. On the one hand, the use of MSNNs provides interpretability, stability, and the possibility of incorporating formal knowledge into the learning process. On the other

The authors are with the Department of Industrial Engineering, University of Trento, Via Sommarive 9, 38123 Povo (TN), Italy (email: gastone.rosatipapini@unitn.it)

This work was supported under the FIS 2 Call, Grant Assignment Decree No. 1236 adopted on 01/08/2023 by the Italian Ministry of University and Research (MUR), for the project FIS-2023-03684 'Structured neural network framework for modeling and control of autonomous systems – Neu4mes', CUP E53C24003800001.

hand, the differentiable integration of planning and control enables continuous adaptation and improvement from data, including fine-tuning on new scenarios and domains. Ultimately, this work advocates for a hybrid paradigm, where modularity is not opposed to learning, but rather embedded within it. By leveraging structured neural representations and differentiable simulation, the proposed framework moves toward autonomous driving systems that are not only scalable and data-efficient, but also robust, interpretable, and capable of generalizing beyond their training distribution.

II. Fully Differentiable Planning Architecture

.. Componenti Schema generale (con i tre blocchi) e idea conclusiva

... Training

The low-level control was implemented as an MSNN steer controller, while a PI controller on the desired acceleration was used for longitudinal control.

A. Vehicle Model

The vehicle MSNN model for lateral dynamics relates the steering input angle to the resulting vehicle curvature. The network's physical structure is derived from the handling diagram (HD), assuming that the understeer factor K_{us} varies linearly with lateral acceleration ($a_y = \rho v_x^2$):

$$\rho(a_x, v_x, \delta) = \frac{\delta}{L + K_{us}(a_x) v_x^2}. \quad (1)$$

Here, ρ denotes the vehicle curvature, δ is the steering angle at the wheels, and L represents the wheelbase (inter-axle distance). The proposed neural network predicts the vehicle curvature at the next time step ρ_{k+1} (4) using a time window of $n = 30$ past steering samples, $\boldsymbol{\delta}_k = [\delta_{k-n+1}, \ldots, \delta_k]$, together with current longitudinal speed v_k and acceleration a_k. The model is structured as the product of a dynamic term (2) which captures the steering-to-curvature relationship through a speed-dependent FIR filter ($\boldsymbol{w}_{\boldsymbol{\delta}}$ are the weights), and a parametric function with parameter A_i (3) that accounts for understeer effects under different longitudinal acceleration conditions. $\phi_{*i}(\cdot)$ denotes triangular fuzzy membership function. The number of channels is $N_v = 8$, $N_a = 5$. The neural model's equations are:

$$D_v(v_k, \boldsymbol{\delta}_k) = \sum_{i=1}^{N_v} \phi_{v_i}(v_k)\, ({\boldsymbol{w}_{\boldsymbol{\delta}}}_i^\top \boldsymbol{\delta}_k), \quad (2)$$

$$F_a(a_k, v_k) = \sum_{i=1}^{N_a} \phi_{a_i}(a_k)\, \frac{1}{1 + A_i v_k^2}, \quad (3)$$

$$\rho_{k+1} = D_v(v_k, \boldsymbol{\delta}_k) F_a(a_k, v_x). \quad (4)$$

B. Lateral Control

The controller receives two future windows of length $n = 30$ for the target velocity $\boldsymbol{v_t}_k$ and curvature $\boldsymbol{\rho_t}_k$, along with a past window of realized steering angles $\boldsymbol{\delta_c}_k = [\delta_{k-n+1}, \ldots, \delta_{k-1}]$ to account for initial steering conditions. A parametric function (6), using parameters A_i learned by the forward model, compensates for understeer effects, while a FIR filter (5) ($\boldsymbol{w}_{\boldsymbol{\rho}}$) captures the vehicle's dynamic response. The predicted steering angle $\tilde{\delta}_k$ (7) is obtained as the sum of the target trajectory contribution and the past steering history.

$$K_t(a_k, \boldsymbol{v_t}_k, \boldsymbol{\rho_t}_k) = \sum_{i=1}^{N_v} \phi_{v_i}(v_k)\, ({\boldsymbol{w}_{\boldsymbol{\rho}}}_i^\top \nabla(a_k, \boldsymbol{v_t}_k, \boldsymbol{\rho_t}_k)) \quad (5)$$

$$\nabla(a_k, \boldsymbol{v_t}_k, \boldsymbol{\rho_t}_k) = \boldsymbol{\rho_t}_k \odot \left(1 + \sum_{i=1}^{N_a} \phi_{a_i}(a_k)\, A_i\, \boldsymbol{v_t}_k^2\right) \quad (6)$$

$$\tilde{\delta}_k = K_t(a_k, v_k, \boldsymbol{v_t}_k, \boldsymbol{\rho_t}_k) + {\boldsymbol{w}_{\boldsymbol{\delta_c}}}^\top \boldsymbol{\delta_c}_k \quad (7)$$

C. Motion Planner

The proposed architecture employs a jerk-optimized motion-planning layer to generate smooth, high-fidelity trajectories. At its core, this layer solves a constrained optimization problem to minimize control effort, defined as $J = \int_0^{t_f} \mathbf{u}(t)^2\, \mathrm{d}t$, where the control input $\mathbf{u}(t)$ corresponds to longitudinal or lateral jerk. By formulating the motion primitives as the result of an Optimal Control Problem (OCP), the architecture can produce solutions that are inherently compliant with the system dynamics and boundary conditions.

While these primitives can be derived through analytical optimality conditions, the framework is designed to be independent of the underlying solver. This allows the motion planning layer to be implemented either through closed-form analytical solutions or via structured neural networks trained to approximate the optimal control manifold.

Regardless of the implementation, the resulting primitives serve as a basis for the decision-making layer. By evaluating a *salience* metric against these candidates, the system measures the quality and robustness of the trajectories relative to the environmental context, ensuring the final control output is both optimal and context-aware.

More specifically, the total salience is computed as a weighted sum of longitudinal and lateral contributions, designed to balance efficiency, comfort, and safety.
The longitudinal component consists of: (i) the maneuver duration from an initial condition to a final one; (ii) the OCP cost $J = \int_0^{t_f} \mathbf{u}(t)^2\, \mathrm{d}t$; and (iii) a velocity constraint penalty. Similarly, the lateral salience accounts for: (i) the time to reach the road boundary; (ii) the time required to re-enter the bounds; and (iii) the initial lateral jerk r_0.

To enable the management of complex environments, an additional term derived from a structured neural network can be integrated into this calculation. This learned component allows the planner to adapt to non-linear environmental features and sophisticated constraints that are difficult to model analytically, effectively refining the decision-making process in real-time.

Following the salience assignment, the *motor cortex* is constructed to provide a topographic organization of the motion space. In this structure, action pairs are arranged onto a two-dimensional grid, where each cell corresponds to a specific combination of longitudinal and lateral jerk, and its value represents the total salience of that maneuver. By placing similar actions in proximity, a structured map

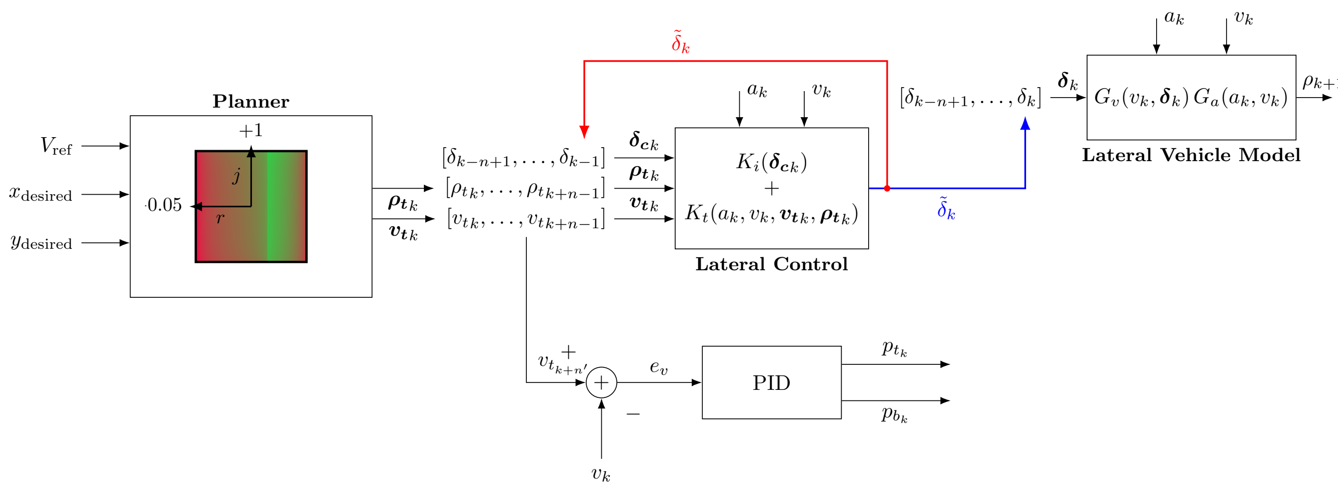

Fig. 1. Block diagram of the proposed pipeline: a minimum-jerk planner selects the optimal primitive from a jerk-based grid (green), given a lookahead target position ($x_{\text{desired}}, y_{\text{desired}}$) and reference speed computed according to the maxixmum road curvature (V_{ref}), and outputs future curvature $\boldsymbol{\rho_{t}}_k$ and velocity $\boldsymbol{v_{t}}_k$ profiles. A PID handles longitudinal control, while steering is generated by a neural network trained in closed loop with a neural vehicle model.

of the available motion space is created. This organization ensures the explainability of the planner, as the relationship between specific control inputs and their associated costs, such as those resulting from road boundaries, can be clearly visualized.

The final action selection is performed through a winner-take-all procedure, which identifies the grid cell with the minimum salience value. This process results in the selection of the most efficient and safe trajectory from the calculated options.

D. *Framework*

....

III. Experimental Results

A. *Experimental Current Setup*

IV. Conclusions and Future Work

This work presented a first experimental validation of a hybrid framework for autonomous driving based on Model Structure Neural Networks, focusing on the integration of motion planning and low-level control within a differentiable architecture. The proposed approach has been evaluated in a high-fidelity commercial simulation environment (IPG CarMaker®), where the planner and the control components are designed to be implemented through structured neural networks. At the current stage, only the low-level control module has been fully implemented using the nnodely framework, demonstrating the feasibility of learning structured neural controllers and deploying them in closed-loop simulations. The planner component, although conceptually integrated within the proposed architecture, remains to be fully developed within the same framework. Future work will focus on extending the nnodely-based implementation to the entire control pipeline, including the motion planning layer. This will enable the realization of a fully unified and differentiable architecture, where all modules are trained within a common framework. Such an approach would allow multi-level training strategies, supporting both independent module optimization and end-to-end fine-tuning across planning and control. Another key research direction concerns online adaptation. Since the proposed controllers are based on neural networks, they naturally enable the possibility of adapting their behavior during operation. Investigating robust and safe online learning mechanisms will be essential to further improve generalization capabilities and allow the system to cope with previously unseen conditions.

Overall, this work represents an initial step toward structured, learnable, and adaptable autonomous driving systems, highlighting the potential of MSNN-based approaches for bridging modular and end-to-end paradigms.

References

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[2] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 164–10 183, 2024.

[3] J. Hu, Y. Wang, S. Cheng, J. Xu, N. Wang, B. Fu, Z. Ning, J. Li, H. Chen, C. Feng *et al.*, "A survey of decision-making and planning methods for self-driving vehicles," *Frontiers in Neurorobotics*, vol. 19, p. 1451923, 2025.

[4] M. Da Lio, D. Bortoluzzi, and G. P. Rosati Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle System Dynamics*, vol. 58, no. 11, pp. 1675–1693, 2020. [Online]. Available: https://doi.org/10.1080/00423114.2019.1638947

[5] M. Piccinini, S. Taddei, M. Larcher, M. Piazza, and F. Biral, "A physics-driven artificial agent for online time-optimal vehicle motion planning and control," *IEEE Access*, vol. 11, pp. 46 344–46 372, 2023.

[6] M. Da Lio, R. Donà, G. P. Rosati Papini, F. Biral, and H. Svensson, "A mental simulation approach for learning neural-network predictive control (in self-driving cars)," *IEEE Access*, vol. 8, pp. 192 041–192 064, 2020.