Disentangled Differentiable Model Predictive Control for Data-efficient and Interpretable Imitation Learning

Jonghak Bae [1] **Taehyung Kim** [2] **Yonghwa Seo** [1] **Jongeun Choi** [1]

Abstract

Efficient imitation of expert behaviors in high-dimensional continuous control remains a fundamental challenge, particularly when balancing physical safety with adaptation to varying task environments. In this paper, we propose a framework that treats expert behavior as a dynamic composition of learnable control primitives—interpretable cost function components within a differentiable Model Predictive Control (MPC) layer. By reformulating imitation learning from a pure black-box regression into a structured grey-box optimization, our model disentangles complex expert strategies into shared strategic bases and a context-aware gating network. This gating mechanism, conditioned via Feature-wise Linear Modulation (FiLM), integrates temporal motion history with environmental context to dynamically modulate control primitive activations, ensuring seamless strategic transitions across varying geometries. We validate our approach on high-speed autonomous racing benchmarks, where the framework demonstrates superior fidelity and transparency. Notably, the architecture enables rapid adaptation to out-of-distribution contexts, achieving near-expert performance on unseen geometries via few-shot refinement with only a single lap of data. These results highlight the efficiency of structured differentiable layers in distilling robust, interpretable decision-making policies from offline datasets.

[1]School of Mechanical Engineering, Yonsei University, Seoul, South Korea [2]Department of Mobility Systems Engineering, Yonsei University, Seoul, South Korea. Correspondence to: Jongeun Choi <jongeunchoi@yonsei.ac.kr>.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.

1. Introduction

Mastering complex tasks, such as autonomous driving or legged locomotion, often requires emulating human experts who inherently balance multi-objective trade-offs among safety, efficiency, and physical limits. Since such expert behaviors are difficult to formalize using manual heuristics, early research focused on Imitation Learning (IL) to capture underlying patterns from demonstrations. To mitigate inherent out-of-distribution (OOD) issues and improve performance, recent frameworks have introduced more structured representations. For instance, DART (Laskey et al., 2017) improves robustness by injecting noise into expert demonstrations, while CompILE (Kipf et al., 2019) adopts a hierarchical approach to decompose demonstrations into interpretable sub-skills. Building on these ideas, more advanced architectures such as Triple-GAIL (Fei et al., 2020) and MILE (Hu et al., 2022) incorporate skill selection mechanisms or latent world models to address high-dimensional tasks. However, as these methods fundamentally rely on model-free reinforcement learning or end-to-end neural networks, they often suffer from limited stability and a lack of safety guarantees, particularly during the early stages of learning or in unseen scenarios. Consequently, achieving both interpretability and rigorous safety remains challenging, highlighting the need for more fundamentally structured control formulations.

To overcome these challenges, integrating IL within the Model Predictive Control (MPC) framework has been extensively explored to enforce physical constraints while mimicking expert behaviors. Various methodologies have been proposed to recover representative cost functions, including bi-level inverse MPC approaches based on trajectory fitting (Ramadan et al., 2018) and analytic inverse MPC methods based on Pontryagin's Maximum Principle (Zhang et al., 2024), spatiotemporal costmaps via inverse reinforcement learning (Lee et al., 2022), and Gaussian Process-based prediction for scheduled cost weights (Bae et al., 2024). While these hybrid approaches enhance stability, they often rely on computationally intensive optimization for tuning (Fröhlich et al., 2022) and lack the structural flexibility to generalize across unseen tasks or diverse environments without substantial retraining (Zhou et al., 2025).

Differentiable optimization layers (Amos & Kolter, 2017; Amos et al., 2018) enable end-to-end learning by propagating gradients through the controller's optimization via KKT conditions and the implicit function theorem (IFT). Unlike decoupled approaches, this differentiable MPC (DMPC) paradigm allows for the simultaneous identification of cost parameters and dynamics—an approach recently extended to noise covariance learning in moving horizon estimation (Jeong & Choi, 2023). In the context of imitation learning, recent evaluations (Acerbo et al., 2023) show that DMPC-based imitation effectively bridges the gap between open-loop learning and closed-loop stability, maintaining safety constraints while ensuring human-like behavior. To further refine performance, sensitivity-based methods like Diff-Tune (Tao et al., 2024) optimize cost functions based on full-trajectory system performance rather than single-step errors. Furthermore, DMPC has been increasingly integrated with high-dimensional sensory inputs. For instance, researchers have explored mapping visual features directly to cost weights (Acerbo et al., 2024), while context-aware encoders (Huang et al., 2023) have been used to dynamically adjust costs in response to complex environmental geometries. Recent state-of-the-art methods further demonstrate DMPC's capability in highly dynamic maneuvers (Jahncke et al., 2026), or providing formal safety guarantees for end-to-end autonomous driving through adaptive barrier-integrated networks (Xiao et al., 2025).

Despite these advancements, existing DMPC frameworks often rely on predefined reference trajectories and primarily focus on weight tuning, limiting interpretability and structural insight into learned behaviors. By treating cost functions as monolithic parameter sets, these models provide little understanding of how specific control strategies emerge. Consequently, adapting to unseen tasks or changing environments typically requires extensive retraining, as fundamental maneuvers and task-specific decision-making are not explicitly disentangled.

To address these limitations, we present a framework that disentangles the control policy into two distinct components: a shared set of strategic bases (MPC cost function primitives) and a task-specific gating network that determines their real-time activations. Unlike monolithic architectures, our approach decomposes complex behaviors into fundamental control primitives learned during a training phase. The gating network dynamically blends these bases, enabling continuous interpolation between strategies to suit varying task geometries and operational requirements. The primary advantage of this disentangled representation is its data efficiency: since core control primitives are encoded within the strategic bases, adapting to new environments or objectives requires only updating the lightweight gating logic. Consequently, the system enables rapid adaptation with minimal data while preserving the safety and physical consistency inherent in the MPC framework. Moreover, the gating activations provide transparency, allowing for the direct interpretation of each strategy's contribution to the final control action.

The key contributions of this work are summarized as follows:

- We propose a differentiable framework that disentangles control policies into shared strategic bases and a gating network, enabling direct interpretation of complex decision-making through sparse activations.
- We introduce a two-stage learning scheme that enables rapid adaptation to novel tasks or environments with minimal demonstration data by leveraging pre-trained control primitives.
- We demonstrate a unified architecture capable of imitating diverse expert styles in highly dynamic scenarios while strictly preserving physical safety constraints and handling limits.

2. Methodology

2.1. Problem Formulation

Consider a discrete-time dynamic system governed by nonlinear dynamics $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$, where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and $\mathbf{u}_k \in \mathbb{R}^{n_u}$ denote the state and control inputs at step k. Our framework adopts a grey-box policy representation, where the agent's decision-making is structured as a parameterized MPC layer. At each time step t, the policy solves a finite-horizon Optimal Control Problem (OCP) with a prediction horizon N to determine the optimal control sequence $\mathbf{U}_t^* = \{\mathbf{u}_{t,0}^*, \dots, \mathbf{u}_{t,N-1}^*\}$:

$$(\mathbf{X}_t^*, \mathbf{U}_t^*) = \underset{\mathbf{X}_t, \mathbf{U}_t}{\arg\min} \quad \sum_{k=0}^{N-1} J(\mathbf{x}_k, \mathbf{u}_k; \mathbf{z}_t) + J(\mathbf{x}_N; \mathbf{z}_t) \tag{1}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \\ & \mathbf{x}_0 = \mathbf{x}_t, \\ & \mathbf{x}_k \in \mathcal{X},\ \mathbf{u}_k \in \mathcal{U}, \\ & k \in \{0, \dots, N-1\}, \end{aligned} \tag{2}$$

where $\mathbf{x}_t$ represents the current state measurement, and $\mathcal{X}, \mathcal{U}$ denote the state and control constraint sets, respectively. Following the receding horizon principle, only the first element of the optimal sequence, $\mathbf{u}_{t,0}^*$, is applied to the system.

Let $\mathcal{D} = \{\zeta_i^{\text{EXP}}\}_i$ be an offline dataset of expert demonstrations. Each trajectory $\zeta_i^{\text{EXP}} = \{(\mathbf{x}_t^{\text{EXP}}, \mathbf{u}_t^{\text{EXP}})\}_{t=0}^{T_i}$ is collected under a specific task configuration $\mathcal{T}_i \sim p(\mathrm{T})$, representing diverse environmental geometries or operational

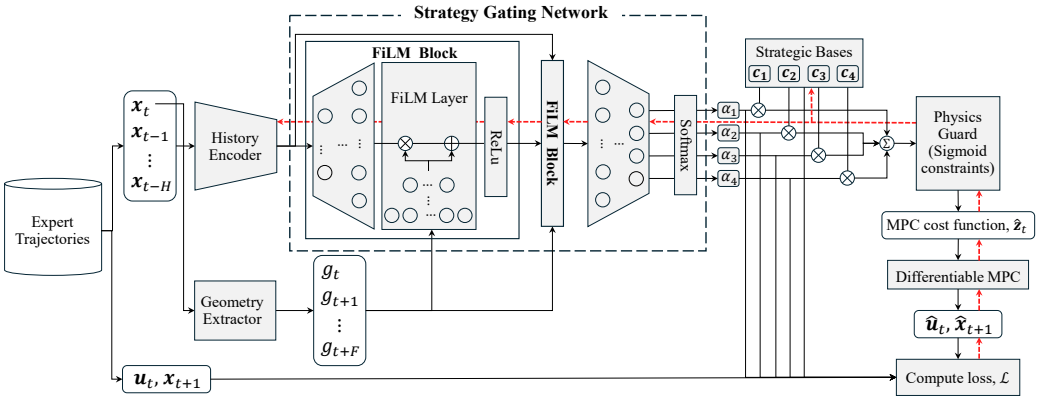


Figure 1. Schematic of the proposed end-to-end pipeline. Expert trajectories are disentangled into strategic primitives via a strategy gating network and a set of learnable strategic bases. The framework integrates spatial-temporal context using FiLM-based modulation to predict context-dependent MPC parameters $\hat{\mathbf{z}}_k$. The Differentiable MPC layer enables direct optimization of the control policy while maintaining physical feasibility through the Physics Guard. Red dashed lines denote the gradient flow during backpropagation.

constraints. The observed behavior $\zeta_{\mathcal{T}}^{\text{EXP}}$ is assumed to be an ϵ-optimal solution to the expert's underlying global objective $\mathcal{J}_{\text{EXP}}(\zeta \mid \mathcal{T})$:

$$\zeta_{\mathcal{T}}^{\text{EXP}} \in \left\{ \zeta : \mathcal{J}_{\text{EXP}}(\zeta \mid \mathcal{T}) \leq \inf_{\tilde{\zeta}} \mathcal{J}_{\text{EXP}}(\tilde{\zeta} \mid \mathcal{T}) + \epsilon \right\}, \quad (3)$$

where $\epsilon \geq 0$ accounts for potential suboptimality in the demonstrations.

Our objective is to find a mapping f_θ that predicts context-dependent parameters $\hat{\mathbf{z}}_t$ given the task configuration $\mathcal{T}$. Optimization of network weights θ aims to minimize the discrepancy between expert actions and the structured policy across diverse configurations:

$$\theta^* := \arg\min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathrm{T})} \mathbb{E}_{(\mathbf{x}_t^{\text{EXP}}, \mathbf{u}_t^{\text{EXP}})\sim\zeta_{\mathcal{T}}^{\text{EXP}}} \left[\mathcal{L}\left(\mathbf{u}_t^{\text{MPC}}, \mathbf{u}_t^{\text{EXP}}\right) \right], \quad (4)$$

where $\mathbf{u}_t^{\text{MPC}}$ is the first control action obtained by solving (1) at state $\mathbf{x}_t^{\text{EXP}}$ with the predicted parameters $\hat{\mathbf{z}}_t$, and $\mathcal{L}$ is a loss function measuring the discrepancy between the MPC and expert actions.

2.2. Learning Framework for Disentangled Decision Primitives

Figure 1 illustrates the end-to-end architecture of the proposed framework, which decomposes complex agent behaviors into weighted combinations of fundamental cost primitives. By adopting a Mixture-of-Experts (MoE) structure, the model predicts context-dependent objective parameters $\mathbf{z}_t$ through a strategy gating network that analyzes spatio-temporal context to determine the optimal blending of these primitives. Implicit differentiation through a differentiable MPC layer enables end-to-end training by backpropagating gradients through the KKT optimality conditions (Amos & Kolter, 2017; Amos et al., 2018). By applying the IFT, the gradient of the imitation loss flows back to the gating network via a single linear system solve, bypassing explicit KKT matrix inversion. This approach allows the model to minimize imitation discrepancy while ensuring the learned policy remains strictly consistent with physical and safety constraints throughout the training process.

The strategy gating network identifies the operational context by processing heterogeneous data streams. A history encoder, utilizing Gated Recurrent Unit (GRU) layers (Chung et al., 2014), compresses the sequence of past H agent states $\mathbf{x}_{t-H:t}$ into a temporal feature vector $\mathbf{h}$ to capture the dynamic trend of the system. Concurrently, a geometry extractor processes the upcoming environmental geometry $g_{t:t+F} \in \mathcal{T}$ over a look-ahead horizon F, providing a spatial preview of the task (e.g., future path curvatures and relative positions). By utilizing relative spatial representations, we ensure that the framework remains invariant to specific global coordinates. Rather than relying on simple concatenation, the framework employs Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) to dynamically condition the temporal history on the spatial context. The extracted geometry features $\phi(g_{t:t+F})$ generate learnable scaling (γ) and shifting (β) parameters to modulate the latent representation $\mathbf{h}$:

$$\text{FiLM}(\mathbf{h}|\phi(g_{t:t+F})) = \gamma(\phi(g_{t:t+F})) \odot \mathbf{h} + \beta(\phi(g_{t:t+F})), \quad (5)$$

where $\odot$ denotes the element-wise product. This spatiotemporal modulation effectively reconfigures the network's latent representation, adapting the agent's motion history to the specific strategic requirements of the upcoming task segment.

The modulated features are processed through linear layers and a softmax function to generate the gating weights

$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]^T$. These weights determine the MPC parameter vector by performing a convex combination of the strategic bases $\mathbf{c}_i \in \mathbb{R}^6$, where the number of bases M is selected to balance model expressivity and computational efficiency. Specifically, the bases $\mathbf{c}_i$ are learnable cost function vectors that serve as the strategic primitives for $\mathbf{z}_t$ in the MPC cost function. Initialized with hand-tuned prior knowledge, these bases are refined alongside the gating network through the DMPC layer. To ensure physical consistency, the combination is followed by a Physics Guard (Chrosniak et al., 2024) mapping to produce the final parameter vector $\mathbf{z}_t$:

$$\hat{\mathbf{z}}_t = \sigma\left(\sum_{i=1}^{M} \alpha_i \mathbf{c}_i\right) \odot (\bar{\Phi} - \underline{\Phi}) + \underline{\Phi}, \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function, and $\bar{\Phi}, \underline{\Phi}$ are the predefined physical bounds for each parameter. This formulation ensures that the predicted objectives remain numerically stable and physically feasible. Upon the convergence of training, the refined bases $\mathbf{c}_i$ serve as fundamental strategic primitives capable of representing a diverse range of driving styles. Since the gating network explicitly determines which strategy basis to activate and to what extent at any given moment, the framework provides high interpretability regarding the controller's decision-making process while ensuring stable and robust imitation.

2.3. Loss Function

The training objective is to minimize the discrepancy between the grey-box policy predictions and expert demonstrations while ensuring the interpretability and diversity of the learned primitives.

Imitation Loss ($\mathcal{L}_{imit}$) ensures robust trajectory reproduction and mitigates compounding errors during closed-loop execution by incorporating both action and state-transition discrepancies:

$$\mathcal{L}_{imit} = \mathbb{E}\left[\|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|^2_{\mathbf{W}_x} + \|\hat{\mathbf{u}}_t - \mathbf{u}_t\|_2^2\right], \tag{7}$$

where $\hat{\mathbf{u}}_t$ and $\hat{\mathbf{x}}_{t+1}$ represent the control action and subsequent state predicted by the differentiable optimization layer, evaluated against expert samples $(\mathbf{x}_{t+1}, \mathbf{u}_t)$. The diagonal matrix $\mathbf{W}_x$ assigns non-zero weights exclusively to the core kinematic variables.

Diversity Loss ($\mathcal{L}_{div}$) prevents the learned decision primitives $\mathbf{c}_i$ from collapsing into redundant representations by imposing an orthogonality constraint. This encourages the primitives to span mutually orthogonal directions in the parameter space, thereby enhancing the model's expressive diversity. Let $\tilde{\mathbf{C}} \in \mathbb{R}^{M \times d}$ be the row-normalized matrix of the primitives. The loss penalizes the deviation of its Gram matrix from the identity $\mathbf{I}_M$:

$$\mathcal{L}_{div} = \left\|\tilde{\mathbf{C}}\tilde{\mathbf{C}}^T - \mathbf{I}_M\right\|_F^2. \tag{8}$$

Sparsity Loss ($\mathcal{L}_{spr}$) ensures high interpretability by encouraging the gating network to perform decisive selection among primitives, approaching a one-hot activation. Maximizing the expected infinity norm of the gating weights $\boldsymbol{\alpha}_t$ promotes this sparsity, explicitly penalizing heavy interpolation between disparate strategies:

$$\mathcal{L}_{spr} = -\mathbb{E}\left[\|\boldsymbol{\alpha}_t\|_\infty\right]. \tag{9}$$

Balance Loss ($\mathcal{L}_{bal}$) mitigates potential mode collapse where the model might rely on a restricted subset of primitives globally. While sparsity encourages decisive selection per sample, maximizing the entropy of the marginal gating distribution $\bar{\boldsymbol{\alpha}} = \mathbb{E}[\boldsymbol{\alpha}_t]$ ensures uniform utilization across the primitive set:

$$\mathcal{L}_{bal} = \sum_{i=1}^{M} \bar{\alpha}_i \log(\bar{\alpha}_i + \epsilon), \tag{10}$$

where ϵ is a small constant for numerical stability. This regularizer ensures that all primitives are globally utilized, preserving the full expressive capacity of the MoE framework.

The total loss $\mathcal{L}_{total}$ is formulated as a weighted sum of the imitation loss and three distinct regularization terms:

$$\mathcal{L}_{total} = \mathcal{L}_{imit} + \lambda_{div}\mathcal{L}_{div} + \lambda_{spr}\mathcal{L}_{spr} + \lambda_{bal}\mathcal{L}_{bal}, \tag{11}$$

where $\lambda_{div}, \lambda_{spr}$, and λ_{bal} are hyperparameters controlling the relative importance of each regularization term.

3. Experimental Setup

In this section, we detail the experimental configuration used to validate our framework. We evaluate its performance in the context of high-speed autonomous racing—a domain that demands both rigorous safety guarantees and the ability to capture diverse, complex expert strategies. We first describe the simulation environment and the underlying MPC formulation. Subsequently, we provide the implementation details of our framework and the baseline methods used for comparison.

3.1. Simulation Environment

We employ a full-scale bicycle model of a Hyundai Ioniq 5 with parameters identified from IPG CarMaker. The state and control vectors are defined as $\mathbf{x} =$

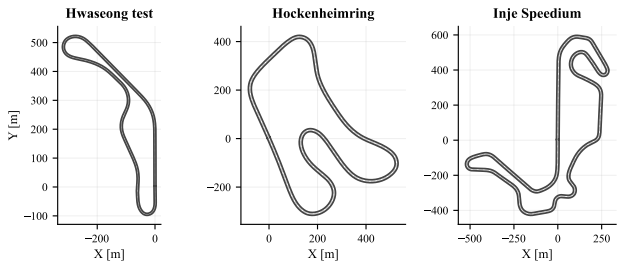


Figure 2. Layouts of the evaluation tracks: Hwaseong test track, Hockenheimring, and Inje Speedium. Black circles represent the starting points, and arrows indicate the driving direction.

$[s, e_c, e_\psi, v_x, v_y, \omega, \delta, \tau]^T$ and $\mathbf{u} = [\Delta\delta, \Delta\tau]^T$, respectively. Details of the vehicle model are provided in Appendix A. The dataset encompasses three tracks with varying spatial characteristics to test the framework's adaptability: Hwaseong test track, a technical circuit characterized by its narrow width and high-frequency tight corners; Hockenheimring (CarMaker built-in), which provides a balanced, mid-scale layout for evaluating standard high-speed maneuvers; and Inje Speedium, the most demanding environment featuring the longest lap length and extreme curvatures that test the vehicle's handling limits, as depicted in Figure 2. To train and evaluate the framework, we generated a diverse dataset of expert trajectories using a path-parametric MPC formulation. We collected data representing distinct driving styles—namely Optimal (Aggressive), Myopic (Short-sighted), and Centerline-following styles—across all evaluation tracks. The detailed procedures for data collection and expert policy configurations are provided in Appendix B.

3.2. MPC Design

As the proposed framework determines control inputs through a DMPC layer, it is essential to develop a robust MPC formulation tailored to the racing environment. The proposed policy is inherently constrained by the expressive capacity of the MPC over the prediction horizon N. The objective function and constraints of the MPC are defined as follows.

Objective Function To enable multi-expert and multi-track imitation, we extend the MPC's expressive capacity by dynamically predicting context-dependent parameters. The surrogate objective function is formulated to balance lateral deviation and velocity tracking:

$$\begin{aligned} J(\mathbf{x_k}, \mathbf{u_k}; \mathbf{z}_t) = &\|v_{x,k} - v_{\text{ref}}\|^2_{q_v} + \|e_{c,k} - e_{c,\text{ref}}\|^2_{q_{e_c}} \\ &+ \|e_{\psi,k}\|^2_{q_{e_\psi}} + \|\omega_k\|^2_{q_\omega} \\ &+ \|\Delta\delta_k\|^2_{r_{\Delta\delta}} + \|\Delta\tau_k\|^2_{r_{\Delta\tau}}, \end{aligned} \tag{12}$$

where the learnable parameter vector $\mathbf{z}_t = [q_v, q_{e_c}, r_{\Delta\delta}, r_{\Delta\tau}, v_{\text{ref}}, e_{c,\text{ref}}]^T \in \mathbb{R}^6$ is dynamically predicted by the network f_θ. Weights q_{e_ψ} and q_ω are kept constant to ensure fundamental directional stability and yaw damping across all driving styles. The terminal cost J_N shares this structure but excludes control increment penalties. By predicting local references $(v_{\text{ref}}, e_{c,\text{ref}})$ alongside their priorities (q_v, q_{e_c}), the network explicitly modulates target racing lines and strategic priorities. Additionally, including $r_{\Delta\delta}$ and $r_{\Delta\tau}$ allows f_θ to adjust control smoothness in real-time. This architecture enables the explicit learning of critical racing components, approximating the expert's implicit long-horizon intent through a highly transparent and interpretable parameterization.

Constraints Different from the adaptive objective function, the constraints remain invariant across all tracks and styles to ensure physical feasibility and safety:

- **Actuator Limits**: δ, τ, and their increments are bounded by mechanical and electrical specifications:

$$\delta_{min} \le \delta \le \delta_{max}, \quad \tau_{min} \le \tau \le \tau_{max}, \tag{13}$$

$$\Delta\delta_{min} \le \Delta\delta \le \Delta\delta_{max}, \quad \Delta\tau_{min} \le \Delta\tau \le \Delta\tau_{max}. \tag{14}$$

- **Safety and Lateral Stability**: To maintain track adherence and prevent loss of grip, we constrain the state space, tire slip angles $\alpha_{f,r}$, and lateral acceleration a_{lat}:

$$\begin{aligned} |e_c| \le e_{c,max}, \quad v_{x,min} \le v_x \le v_{x,max}, \\ |\alpha_{f,r}| \le \alpha_{max}, \quad |a_{lat}| \le a_{max}. \end{aligned} \tag{15}$$

Nonlinear stability constraints are implemented as soft constraints using slack variables to ensure recursive feasibility. Furthermore, each state and control variable is normalized relative to its maximum observed value. This normalization improves numerical stability and ensures consistent track boundary enforcement across varying track widths, regardless of the physical scales.

3.3. Implementation Details

The implementation of the DMPC layer leverages the efficient solver framework proposed in (Frey et al., 2025), which is built upon acados (Verschueren et al., 2022). Unlike previous methods limited to convex formulations, this solver handles general NLPs using a Sequential Quadratic Programming (SQP) approach. By differentiating the smoothed optimality conditions of the underlying Interior Point Method (IPM), we can efficiently compute adjoint sensitivities even in the presence of nonlinear inequality constraints. Furthermore, because track widths vary across different racing circuits, we instantiate a dedicated DMPC module for each track to maintain consistent spatial normalization.

Our network is implemented in PyTorch (Paszke et al., 2019). A history encoder (2-layer GRU, hidden dimension 64, dropout 0.1) processes temporal history ($H = 50$) into features $\mathbf{h} \in \mathbb{R}^{64}$. In addition, a geometry extractor captures the upcoming track geometry by representing it as a 150m lookahead curvature profile ($F = 150$, at 1m intervals). This representation provides a compact and informative description of the track layout relevant for vehicle control. The extracted profile is then encoded by a two-layer MLP into a spatial feature vector $\phi \in \mathbb{R}^{64}$, which is subsequently used for conditioning in a FiLM block. These are refined through two sequential FiLM blocks: the first applies dual modulations with residual connections in $\mathbb{R}^{128}$, and the second serves as a $\mathbb{R}^{64}$ bottleneck. Scaling (γ) and shifting (β) parameters are dynamically generated from ϕ via auxiliary MLPs (dimension 64). A final linear layer maps these features to M gating weights. The M strategic bases are initialized with hand-tuned parameters. The number of strategic bases M was set to 4, which was empirically determined to provide the best imitation performance across test data. To ensure physical feasibility, the Physics Guard enforces predefined bounds on the MPC parameters via Eq. (6): cost weights $(q_v, q_{e_c}, r_{\Delta\delta}, r_{\Delta\tau})$ are bounded within $[10^{-2}, 10^2]$ using logarithmic scaling, while reference targets $(v_{\text{ref}}, e_{c,\text{ref}})$ are bounded within $[0.1, 1]$, and $[-1, 1]$ using linear scaling, respectively. The entire framework is trained end-to-end using the Adam optimizer with a learning rate of 10^{-4} and a batch size of 1024. The weighting matrix $\mathbf{W}_x$ in Equation (7) is defined as a diagonal matrix corresponding to the states (v_x, v_y, ω).

To benchmark the performance of the proposed framework, we compare it against several baseline methods, including two approaches using static weights, namely Hand-MPC and DiffTune-MPC, as well as two approaches that predict step-varying weights during driving: Continuous-Concat MPC (CC-MPC) and Continuous-Backbone MPC (CB-MPC). These baselines were selected to assess the individual contributions of the proposed disentangled representation and the Differentiable MPC (DMPC) layer. Detailed descriptions of the baseline architectures and their respective training setups are provided in Appendix C.

4. Results

This section evaluates our framework against baseline methods and highlights its unique strategic interpretability. We further demonstrate the framework's high-fidelity imitation and efficiency through track adaptation results across diverse environments.

4.1. Comparative Analysis

To evaluate the imitation performance and generalizability of the proposed framework, we compare with baseline methods on the Hwaseong test track, training separate models for three driving styles and conducting closed-loop evaluations from the same initial state. Imitation performance is assessed using Mean Absolute Error (MAE) for absolute accuracy and Mean Z-score for statistical consistency. While MAE measures deviation from the expert mean trajectory μ_{EXP}, it penalizes natural variability. To address this, we use the Mean Z-score, defined as $Z = |x_{\text{pred}} - \mu_{\text{EXP}}|/\sigma_{\text{EXP}}$, which normalizes errors by the expert variance. A Z-score below 1.0 indicates predictions within the expert's natural distribution. Safety is evaluated using Track-Out Percentage (TOP), computed spatially to avoid bias from varying speeds. By sampling lateral deviation $e_c(s)$ at 1-meter intervals along the track, TOP is defined as:

$$\text{TOP} = \frac{100}{N_s} \sum_{i=1}^{N_s} \mathbb{I}\big(|e_c(s_i)| > B\big), \tag{16}$$

where N_s is the number of spatial samples and B is the track half-width.

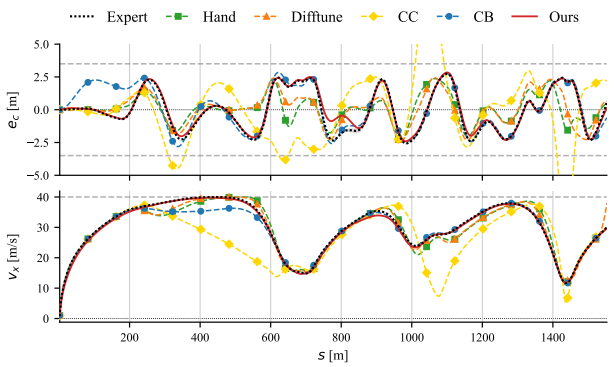


Figure 3. Comparison of tracking performance against expert trajectories. The plots show lateral error e_c (top) and longitudinal velocity v_x (bottom) over track progress s. Light dashed gray lines indicate track boundaries and velocity limits.

4.1.1. Imitation Performance

Unlike static offline testing that measures single-step accuracy, closed-loop evaluation exposes the policy to sequential dynamics and compounding errors, providing a more rigorous assessment of driving performance and robustness to OOD states. Table 1 summarizes the results across all styles, and Figure 3 presents a representative Optimal driving case, comparing e_c and v_x with the expert.

Hand-MPC shows noticeable degradation, particularly in the Optimal and Centerline styles. Although lap time differences remain moderate (+1.45s and +3.10s), tracking errors are significantly higher due to the mismatch in prediction horizons. While it performs comparably in the Myopic style (+0.08s), it fails to reproduce long-term planning behaviors.

Table 1. Comparative evaluation of driving styles on the Hwaseong test track.

Model	Style	Lap Time [s]	State Tracking Errors w.r.t. Expert Mean Trajectory (μ_{exp})										TOP [%]
			Δe_c		Δv_x		Δv_y		$\Delta \omega$		Average		
		(Diff to Ref.)	MAE	Z	MAE	Z	MAE	Z	MAE	Z	MAE	Z	
Hand	Optimal	58.40 (+1.45)	1.042	0.654	1.425	0.156	0.126	0.225	0.023	0.102	0.654	0.284	0.00
	Myopic	59.25 (+0.08)	0.405	0.277	0.284	0.031	0.062	0.121	0.012	0.054	0.191	0.121	0.00
	Centerline	63.15 (+3.10)	0.242	0.596	1.902	0.233	0.098	0.240	0.020	0.088	0.566	0.289	0.00
DiffTune	Optimal	57.10 (+0.15)	0.952	0.597	1.074	0.118	0.114	0.202	0.028	0.123	0.542	0.260	0.00
	Myopic	59.40 (+0.23)	0.241	0.164	0.430	0.047	0.044	0.086	0.011	0.047	0.181	0.086	0.00
	Centerline	60.05 (0.00)	0.476	1.169	1.473	0.181	0.091	0.224	0.020	0.088	0.515	0.415	0.00
CC	Optimal	72.00 (+15.05)	2.441	1.530	5.240	0.573	0.248	0.441	0.046	0.199	1.994	0.686	13.36
	Myopic	64.75 (+5.58)	1.837	1.255	2.140	0.235	0.196	0.385	0.043	0.188	1.054	0.516	18.40
	Centerline	75.90 (+15.85)	2.473	6.075	7.066	0.867	0.311	0.762	0.072	0.317	2.480	2.005	9.99
CB	Optimal	57.85 (+0.90)	0.467	0.302	0.959	0.105	0.059	0.105	0.011	0.048	0.374	0.140	0.00
	Myopic	81.05 (+21.88)	5.051	3.451	21.608	2.377	0.729	1.429	0.232	1.009	6.905	2.067	10.38
	Centerline	61.80 (+1.75)	0.595	1.461	1.098	0.135	0.066	0.162	0.014	0.064	0.443	0.455	0.00
Ours	Optimal	58.00 (+1.05)	**0.175**	**0.113**	**0.445**	**0.049**	**0.041**	**0.074**	**0.007**	**0.031**	**0.167**	**0.067**	**0.00**
	Myopic	60.35 (+1.18)	**0.121**	**0.083**	**0.557**	**0.061**	**0.032**	**0.062**	**0.007**	**0.030**	**0.179**	**0.059**	**0.00**
	Centerline	61.45 (+1.40)	**0.088**	**0.216**	**0.540**	**0.066**	**0.024**	**0.058**	**0.008**	**0.037**	**0.165**	**0.094**	**0.00**

DiffTune-MPC slightly improves performance by optimizing the hand-tuned cost weights. As shown in Table 1, the state tracking errors are reduced across all styles. However, the improvement remains limited because DiffTune-MPC retains the same control structure and short prediction horizon. As a result, despite achieving competitive lap times, the controller still exhibits relatively large tracking errors, indicating a fundamental inability to faithfully recover the expert's driving strategy.

CC-MPC exhibits the most severe degradation among all baselines. By directly concatenating features without structural constraints, it fails to predict parameters aligned with expert strategies, resulting in large tracking errors, including high velocity error in the Centerline style (v_x MAE: 7.066) and frequent track-outs in the Myopic style (TOP: 18.40%). These failures indicate loss of control stability due to accumulated errors, confirming that naive feature concatenation is insufficient to capture the discrete strategic transitions in expert demonstrations.

CB-MPC improves performance in moderate styles (Optimal and Centerline) but fails in the highly dynamic Myopic case, producing large lap time delays ($+21.88$s) and frequent track-outs (TOP: 10.38%). Although its offline accuracy is comparable to ours, its closed-loop performance degrades significantly, indicating that purely black-box networks lack the structural regularization needed for stable parameter prediction and are prone to compounding errors and OOD states.

In contrast, our method consistently achieves the best performance across all styles. By combining FiLM-conditioned gating with explicit strategic bases, it produces stable parameter predictions and effectively disentangles diverse driving strategies. The model achieves near-expert lap times (within 1.4s) with zero track-outs across all styles, while maintaining mean Z-scores below 1.0, indicating faithful reproduction of expert behaviors. Overall, these results show that static MPC tuning or unstructured predictors are insufficient; structured modeling of discrete strategies is essential for stable and accurate imitation in closed-loop control.

Table 2. Semantic interpretation of the learned strategy bases.

Basis	Parameters & Semantic Interpretation
0	q_v: 59.12, q_{e_c}: 52.51, $r_{\Delta\tau}$: 37.73, $r_{\Delta\delta}$: 19.92 v_{ref}: 0.41, $e_{c,\text{ref}}$: +0.86
	Maintains strict path tracking on the left lane. Balances speed and steering control. $\Rightarrow$ **Left-Biased / Balanced**
1	q_v: 0.08, q_{e_c}: 19.92, $r_{\Delta\tau}$: 0.59, $r_{\Delta\delta}$: 1.07 v_{ref}: 0.74, $e_{c,\text{ref}}$: -0.03
	Dominant lateral weight ensures strict center tracking regardless of speed. $\Rightarrow$ **Center-Keeping**
2	q_v: 96.71, q_{e_c}: 0.89, $r_{\Delta\tau}$: 28.77, $r_{\Delta\delta}$: 1.94 v_{ref}: 1.00, $e_{c,\text{ref}}$: +0.50
	Combines maximum target speed with the highest velocity weight for time-optimal performance. $\Rightarrow$ **Time-Attack / Aggressive**
3	q_v: 2.94, q_{e_c}: 6.42, $r_{\Delta\tau}$: 5.48, $r_{\Delta\delta}$: 42.86 v_{ref}: 0.54, $e_{c,\text{ref}}$: -0.84
	Prioritizes steering smoothness while keeping to the right lane for maximum stability. $\Rightarrow$ **Right-Biased / Comfort**

4.1.2. Interpretability

Different from conventional black-box baselines, our framework learns discrete strategic bases with a context-aware gating network, providing inherent interpretability. To relate this structure to driving behavior, we analyze the learned bases $\mathbf{c}_i$ for the Optimal style in the physical domain. Table 2 summarizes the learned bases and their semantic interpretations.

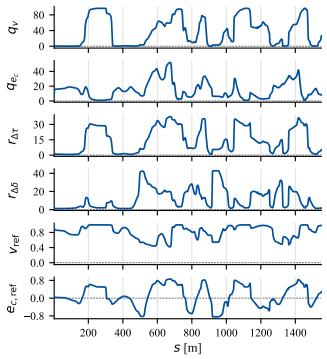
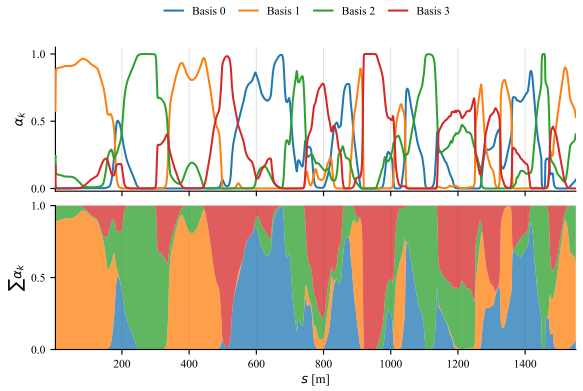


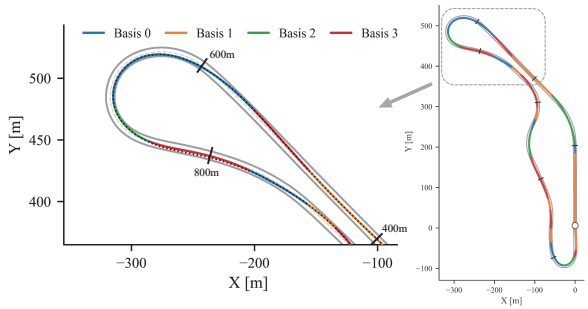
Figure 4. Profiles of the predicted MPC parameter vector $\hat{\mathbf{z}}_k$ entering the DMPC layer, showing the modulation of six parameters over track progress s.

The disentangled strategies show three core axes: control intensity, spatial preference, and stability. First, the model controls aggressiveness by scaling the velocity weight q_v from 0.08 (Basis 1) to 96.71 (Basis 2), covering the range from passive tracking to aggressive driving. Second, it creates spatial diversity by varying the lateral reference $e_{c,\text{ref}}$ between -0.84 (Basis 3) and $+0.86$ (Basis 0), reflecting left/right lane preferences across the track width. Finally, the model enhances smoothness; in Basis 3, the steering rate penalty $r_{\Delta\delta}$ increases to 42.86, approximately 20–40 times higher than in other weights, penalizing abrupt steering. Overall, the proposed Diversity Loss disentangles key driving factors—speed, path, and smoothness—into physically meaningful bases.

Our method employs a gating network to dynamically blend disentangled strategies into final MPC parameters. While Figure 4 shows the synthesized parameters $\hat{\mathbf{z}}k$ over arc-length s, Figure 5a presents the corresponding activation weights $\boldsymbol{\alpha}$. However, relying solely on the synthesized parameters, which share the same format as CB-MPC and CC-MPC, does not provide clear interpretability. While interpretation may be possible when only a subset of parameters varies and others remain stable, in regions where all parameters fluctuate significantly (e.g., $s = 400$–1000m), inferring the underlying decision-making rationale becomes challenging. Our method overcomes this through its sparse, convex gating mechanism ($\sum \alpha_i = 1$). As shown in Figure 5a, the sparsity loss ensures that a single basis $\mathbf{c}_i$ distinctly dominates at any given s, making the agent's intention immediately interpretable. Figure 5b plots the dominant bases ($\alpha_i > 0.5$) on a 2D map, clearly illustrating this advantage. In the U-shaped corner ($s \in [400, 1000]$m), the trajectory breaks down into a clear tactical sequence: entering with Basis 1 (Center-Keeping), shifting to Basis 3 (Right-Biased) for a wider entry, clipping the apex with Basis 0 (Left-Biased), and accelerating out with Basis 2 (Time-Attack). Thus, $\boldsymbol{\alpha}$ not only captures complex maneuvers but also provides a human-interpretable decomposition beyond simple imitation.



(a) Profiles of strategy gating weights α_k over track progress s.



(b) 2D track map colored by dominant strategic bases ($\alpha_i > 0.5$).

Figure 5. Interpretability analysis of the framework. (a) Dynamic blending of strategic bases. (b) Spatial distribution of bases on the track. The black dotted line represents the expert trajectory.

4.2. Context Adaptation Results

The structural disentanglement of strategic bases and the context-aware gating network enables efficient track adaptation. By selecting strategies based on future track layout, a single model can learn multiple track styles jointly. Moreover, previously learned bases can be transferred to adapt to unseen tracks with minimal fine-tuning data. We evaluate three learning setups across three tracks to determine the most efficient learning strategy.

- **Single-Track Training:** A model is trained and evaluated on a single track (3 laps), serving as a track-specific baseline.
- **Joint Training:** A unified model is trained on multiple tracks (3 laps per track) to assess its ability to handle diverse layouts within a single network.
- **Adaptation:** A model pre-trained on two tracks (6 laps) is fine-tuned on an unseen track using a single lap. The history encoder and strategic bases are frozen, and only the gating network is updated; the diversity loss

Table 3. Quantitative comparison of driving performance grouped by Test Track.

Test Track	Method	Train Track	Lap Time [s] (Diff to Expert)	Δe_c MAE	Δe_c Z	Δv_x MAE	Δv_x Z	Δv_y MAE	Δv_y Z	$\Delta \omega$ MAE	$\Delta \omega$ Z	Avg MAE	Avg Z
Hwaseong test (H)	Single	H	58.00 (+1.05)	0.167	0.105	0.442	0.048	0.042	0.075	0.007	0.031	0.165	0.065
	Joint	H + R	57.40 (+0.45)	**0.086**	**0.054**	0.333	0.036	0.036	0.065	0.007	0.029	**0.115**	**0.046**
	Joint	H + I	59.90 (+2.95)	0.157	0.098	0.782	0.086	0.040	0.071	0.008	0.033	0.247	0.072
	Joint	H + R + I	63.65 (+6.70)	0.155	0.097	0.652	0.071	0.027	0.047	**0.005**	**0.023**	0.210	0.060
	Adaptation	R + I	**57.35 (+0.40)**	0.114	0.072	**0.320**	**0.035**	**0.027**	**0.047**	0.007	0.029	0.117	0.046
Hockenheimring (R)	Single	R	90.25 (+4.63)	0.114	0.064	0.441	0.056	0.030	0.048	**0.005**	**0.021**	0.147	0.047
	Joint	H + R	86.65 (+1.03)	0.225	0.127	0.540	0.068	0.070	0.114	0.010	0.049	0.211	0.089
	Joint	R + I	89.90 (+4.28)	0.179	0.101	0.525	0.066	0.034	0.055	**0.005**	**0.026**	0.186	0.062
	Joint	H + R + I	92.55 (+6.93)	0.170	0.096	0.578	0.073	0.048	0.078	0.007	0.034	0.201	0.070
	Adaptation	H + I	**85.95 (+0.33)**	**0.081**	**0.046**	**0.296**	**0.037**	**0.029**	**0.047**	**0.005**	0.023	**0.103**	**0.038**
Inje Speedium (I)	Single	I	137.80 (+3.72)	0.295	0.134	0.729	0.094	0.068	0.102	0.013	0.053	0.276	0.096
	Joint	H + I	138.95 (+4.87)	0.229	0.104	0.712	0.092	0.067	0.101	0.013	0.052	0.255	0.087
	Joint	R + I	140.40 (+6.32)	**0.191**	**0.087**	**0.588**	**0.076**	**0.053**	**0.080**	**0.010**	**0.040**	**0.211**	**0.071**
	Joint	H + R + I	142.20 (+8.12)	0.214	0.098	0.778	0.100	0.064	0.096	0.011	0.046	0.267	0.085
	Adaptation	H + R	**137.65 (+3.57)**	0.329	0.150	0.603	0.078	0.089	0.133	0.019	0.076	0.260	0.109

is removed, and training relies on imitation, sparsity, and balance losses.

Table 3 summarizes the quantitative results across the three learning setups. All test lap times are benchmarked against expert driving, with baseline lap times (Mean $\pm$ Std) of 56.95 ± 0.12s (H), 85.62 ± 0.18s (R), and 134.08 ± 0.69s (I), corresponding to the Hwaseong test, Hockenheimring, and Inje Speedium, respectively.

In the joint training setup, learning compatible tracks can improve imitation fidelity. For example, the H + R model achieves better lateral tracking on H (e_c MAE: 0.086) than the single-track baseline (0.167). However, when trained on all three tracks (H + R + I), performance degrades due to capacity limitations, leading to higher tracking errors and significant lap time delays (e.g., +8.12s on I).

The Adaptation setup overcomes these limits, yielding the most striking results. On R, the adapted model (H + I) achieves the best performance across nearly all metrics, including the tightest state imitation (e_c MAE: 0.081, v_x MAE: 0.296) and a negligible lap time difference of only +0.33s, compared to the Single model's +4.63s. Similarly, adapting R + I to the H achieves the best velocity tracking (v_x MAE: 0.320) and the fastest lap time (+0.40s). For the complex I track, Joint training (R + I) achieves superior state-tracking fidelity by directly internalizing the intricate layout during training. However, when considering system scalability, the Adaptation paradigm offers a more practical solution. Our method demonstrates that a pre-trained model can be efficiently extended to novel environments by fine-tuning only the gating network, maintaining competitive performance (+3.57s) without the prohibitive cost of full joint retraining. Figure 6 shows the state trajectories (e_c, v_x) on unseen tracks after adaptation, demonstrating that the proposed framework achieves high-fidelity imitation with minimal data.

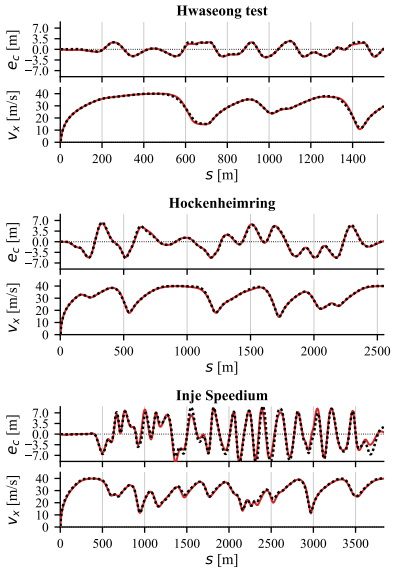


Figure 6. Multi-track validation of driving style imitation. The plots show e_c and v_x over track progress for each track. The solid red line represents our framework, and the dotted black line represents the expert trajectory.

5. Conclusion

We introduced a differentiable framework that reformulates imitation learning into a structured grey-box optimization. By decomposing complex policies into shared strategic bases and a context-aware gating network, our approach captures expert decision-making as a dynamic composition of learnable primitives. Experimental results in limit-handling scenarios demonstrate that the framework achieves superior imitation fidelity and structural interpretability. Furthermore, the disentangled representation enables rapid cross-context adaptation, achieving near-expert performance in novel environments through few-shot refinement with minimal data. These findings highlight structured differentiable layers as a robust, scalable paradigm for distilling interpretable policies from offline datasets. While validated in

autonomous racing, the architecture is applicable to various continuous control tasks where safety and adaptability are paramount. Future work will extend this framework to multi-agent interactions and diverse robotic platforms.

References

Acerbo, F. S., Swevers, J., Tuytelaars, T., and Son, T. D. Evaluation of mpc-based imitation learning for human-like autonomous driving. *IFAC-PapersOnLine*, 56(2): 4871–4876, 2023.

Acerbo, F. S., Swevers, J., Tuytelaars, T., and Son, T. D. Driving from vision through differentiable optimal control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2824–2831. IEEE, 2024.

Amos, B. and Kolter, J. Z. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017.

Amos, B., Jimenez, I., Sacks, J., Boots, B., and Kolter, J. Z. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.

Bae, J., Lim, J., Suh, B., Lee, J., Ryu, K., Kim, J., and Choi, J. Learning expert-level racing strategies via scheduled cost functions in model predictive control. *IEEE Transactions on Intelligent Vehicles*, 2024.

Bakker, E., Nyborg, L., and Pacejka, H. B. Tyre modelling for use in vehicle dynamics studies. Technical report, SAE technical paper, 1987.

Chrosniak, J., Ning, J., and Behl, M. Deep dynamics: Vehicle dynamics modeling with a physics-constrained neural network for autonomous racing. *IEEE Robotics and Automation Letters*, 9(6):5292–5297, 2024.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Fei, C., Wang, B., Zhuang, Y., Zhang, Z., Hao, J., Zhang, H., Ji, X., and Liu, W. Triple-gail: a multi-modal imitation learning framework with generative adversarial nets. *arXiv preprint arXiv:2005.10622*, 2020.

Frey, J., BaumgÃĪrtner, K., Frison, G., Reinhardt, D., Hoffmann, J., Fichtner, L., Gros, S., and Diehl, M. Differentiable nonlinear model predictive control. *arXiv preprint arXiv:2505.01353*, 2025.

Fröhlich, L. P., Küttel, C., Arcari, E., Hewing, L., Zeilinger, M. N., and Carron, A. Contextual tuning of model predictive control for autonomous racing. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10555–10562. IEEE, 2022.

Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., and Shotton, J. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022.

Huang, Z., Liu, H., Wu, J., and Lv, C. Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. *IEEE transactions on neural networks and learning systems*, 35(11):15222–15236, 2023.

Jahncke, F., Zarrouki, B., Piccinini, M., D'sa, J., Isele, D., Bae, S., and Betz, J. Differentiable weights-varying nonlinear mpc via gradient-based policy learning: An autonomous vehicle guidance example. *IEEE Robotics and Automation Letters*, 2026.

Jeong, S. and Choi, J. Differentiable moving horizon estimation for vehicle kinematics via learning covariance matrices. *IEEE Transactions on Intelligent Vehicles*, 9(9): 5955–5969, 2023.

Kipf, T., Li, Y., Dai, H., Zambaldi, V., Sanchez-Gonzalez, A., Grefenstette, E., Kohli, P., and Battaglia, P. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Kloeser, D., Schoels, T., Sartor, T., Zanelli, A., Prison, G., and Diehl, M. Nmpc for racing using a singularity-free path-parametric model with obstacle avoidance. *IFAC-PapersOnLine*, 53(2):14324–14329, 2020.

Laskey, M., Lee, J., Fox, R., Dragan, A., and Goldberg, K. Dart: Noise injection for robust imitation learning. In *Conference on robot learning*, pp. 143–156. PMLR, 2017.

Lee, K., Isele, D., Theodorou, E. A., and Bae, S. Spatiotemporal costmap inference for mpc via deep inverse reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2):3194–3201, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ramadan, A., Choi, J., Radcliffe, C. J., Popovich, J. M., and Reeves, N. P. Inferring control intent during seated balance using inverse model predictive control. *IEEE robotics and automation letters*, 4(2):224–230, 2018.

Ryu, K., Kim, J., Han, J., Bae, J., Suh, B., Lim, J., and Choi, J. An online guide system for improving driving skills on the race track: Visual feedback approach. In *International Conference on Human-Computer Interaction*, pp. 275–282. Springer, 2024.

Tao, R., Cheng, S., Wang, X., Wang, S., and Hovakimyan, N. Difftune-mpc: Closed-loop learning for model predictive control. *IEEE Robotics and Automation Letters*, 9(8): 7294–7301, 2024.

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., Duijkeren, N. v., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., and Diehl, M. acados—a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 14(1):147–183, 2022.

Xiao, W., Wang, T.-H., Gan, C., and Rus, D. Abnet: Adaptive explicit-barrier net for safe and scalable robot learning. In *Forty-second International Conference on Machine Learning*, 2025.

Zhang, F., Duan, J., Xu, H., Chen, H., Liu, H., Nie, S., and Li, S. E. Inverse model predictive control: Learning optimal control cost functions for mpc. *IEEE Transactions on Industrial Informatics*, 20(12):13644–13655, 2024.

Zhou, B., Hu, C., Zeng, J., Li, Z., Betz, J., Xie, L., and Su, H. Adaptive learning-based model predictive control strategy for drift vehicles. *Robotics and Autonomous Systems*, 188:104941, 2025.

A. Vehicle Dynamics

To capture the track-relative motion essential for autonomous racing, we employ a dynamic bicycle model formulated in a curvilinear (Frenet) coordinate system. The state vector is defined as $\mathbf{x} = [s, e_c, e_\psi, v_x, v_y, \omega, \delta, \tau]^T$, where s, e_c, and e_ψ denote the arc-length progress, lateral distance, and heading error relative to the centerline, respectively. Additionally, v_x, v_y, and ω represent the longitudinal velocity, lateral velocity, and yaw rate. The front wheel angle and motor torque are denoted by δ and τ, respectively, and the control input is defined as $\mathbf{u} = [\Delta\delta, \Delta\tau]^T$. The bicycle model is governed by:

$$\begin{aligned}
\dot{s} &= \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1 - \kappa e_c}, \\
\dot{e}_c &= v_x \sin e_\psi + v_y \cos e_\psi, \\
\dot{e}_\psi &= \omega - \kappa \dot{s}, \\
\dot{v}_x &= \frac{F_{rx} + F_{fx}\cos\delta - F_{fy}\sin\delta}{m} + v_y\omega, \\
\dot{v}_y &= \frac{F_{ry} + F_{fx}\sin\delta + F_{fy}\cos\delta}{m} - v_x\omega, \\
\dot{\omega} &= \frac{F_{fx} l_f \sin\delta + F_{fy} l_f \cos\delta - F_{ry} l_r}{I_z},
\end{aligned} \tag{17}$$

where m is the vehicle mass, I_z is the yaw inertia, l_f and l_r are the distances from the center of gravity to the front and rear axles, and κ is the track curvature at a given s.

The longitudinal tire forces are modeled using a simplified torque-based formulation (Bae et al., 2024), which captures front braking in conjunction with rear-wheel drive:

$$\begin{aligned}
F_{fx} &= r_1 \frac{\min(\tau, 0)}{r_w} - C_{rf}\,\mu\, mg, \\
F_{rx} &= \frac{\max(\tau, 0)}{r_w} + r_2 \frac{\min(\tau, 0)}{r_w} - C_{rr}\,\mu\, mg,
\end{aligned}$$

where r_1 and r_2 denote the front and rear brake distribution ratios, C_{rf} and C_{rr} are the rolling resistance coefficients, r_w is the wheel radius, μ is the tire–road friction coefficient, and g is the gravitational acceleration.

The lateral tire forces are described by a simplified Pacejka Magic Formula (Bakker et al., 1987):

$$F_{iy} = D_i \sin\Big(C_i \arctan\big(B_i \alpha_i\big)\Big),$$

where $i \in \{f, r\}$ and B_i, C_i, and D_i denote the stiffness, shape, and peak factors, respectively. The corresponding front and rear slip angles are defined as

$$\begin{aligned}
\alpha_f &= \tan^{-1}\left(\frac{v_y + l_f\omega}{v_x}\right) - \delta \\
\alpha_r &= \tan^{-1}\left(\frac{v_y - l_r\omega}{v_x}\right).
\end{aligned}$$

B. Expert Data Generation

To evaluate the network's capacity to adapt to various driving behaviors, we constructed a dataset encompassing diverse expert styles. To explicitly generate trajectories with distinct driving characteristics, we formulated a Path-Parametric MPC (PP-MPC) based on (Kloeser et al., 2020; Ryu et al., 2024) as the expert controller. The objective function of the expert PP-MPC is defined as:

$$\begin{aligned}
J(\mathbf{x}_k, \mathbf{u}_k; \mathbf{z}_t) = &\|s_{x,k} - s_{ref,k}\|^2_{q_s} + \|e_{c,k}\|^2_{q_{e_c}} + \|e_{\psi,k}\|^2_{q_{e_\psi}} \\
&+ \|\omega_k\|^2_{q_\omega} + \|\tau_k\|^2_{r_\tau} + \|\delta_k\|^2_{r_\delta} \\
&+ \|\Delta\tau_k\|^2_{r_{\Delta\tau}} + \|\Delta\delta_k\|^2_{r_{\Delta\delta}},
\end{aligned} \tag{18}$$

Table 4. Parameter configurations for data generation and our method.

Parameter	Optimal	Myopic	Centerline	Ours
Sampling time	0.05	0.05	0.05	0.05
Prediction Horizon	100	50	100	50
q_s	$\{50, 60, 70\}$	$\{50, 60, 70\}$	10	-
q_{e_c}	10^{-1}	10^{-1}	$\{30, 40, 50\}$	**Learned**
q_{e_ψ}, q_ω	$10^{-1}, 10^{-3}$	$10^{-1}, 10^{-3}$	$10^{-1}, 10^{-3}$	$10^{-1}, 10^{-3}$
r_τ, r_δ	$10^{-3}, 10^{-1}$	$10^{-3}, 10^{-1}$	$10^{-3}, 10^{-1}$	-
$r_{\Delta\tau}, r_{\Delta\delta}$	10^{1}	10^{1}	10^{1}	**Learned**
q_v, v_{ref}	-	-	-	**Learned**
$e_{c,\text{ref}}$	-	-	-	**Learned**

where $s_{x,k}$ is the arc-length along the track centerline, and $s_{ref,k} = s_0 + k\Delta s$ denotes the reference path progress at prediction step k, with s_0 being the initial progress. The spatial increment Δs is fixed at 2 m, which inherently bounds the vehicle's maximum target speed to 40 m/s under the 0.05 s sampling time. The terms $\|\tau_k\|^2_{r_\tau} + \|\delta_k\|^2_{r_\delta}$ are included to differentiate the expert's structure from Eq.(12). To generate diverse driving styles, we define three expert behaviors by varying the prediction horizon and cost weights of the PP-MPC, as summarized in Table 4. Specifically, we consider: 1) Optimal driving, using a long horizon for high-speed, globally optimal trajectories; 2) Myopic driving, using a short horizon for fast but locally focused maneuvers; and 3) Centerline driving, combining a long horizon with strong lateral penalties to maintain centerline adherence. Trajectories are generated by modulating three cost weights. Only three laps per style are collected, which is sufficient for robust imitation. The dataset is split into training, validation, and test sets with a 6:2:2 ratio.

C. Baseline Methods

We evaluate the performance with 4 baseline methods which share the same MPC formulation and prediction horizon ($N = 50$) as our framework.

Static Hand-tuned MPC (Hand-MPC) represents the conventional approach using fixed, manually tuned cost parameters for each style: optimal $\{60, 20, 10, 10, 1, 0\}$, myopic $\{60, 0.1, 10, 10, 1, 0\}$, and centerline $\{10, 40, 10, 10, 1, 0\}$, defined for $\{q_v, q_{e_c}, r_{\Delta\delta}, r_{\Delta\tau}, v_{\text{ref}}, e_{c,\text{ref}}\}$. This highlights the necessity of dynamic adaptation over static tuning.

DiffTune-MPC (Tao et al., 2024) optimizes the tracking weights of Hand-MPC via closed-loop parameter tuning. Initialized with Hand-MPC weights, it updates parameters using gradient information using only the imitation loss (7) across multiple demonstration laps. This evaluates the benefit of context-aware adaptation over globally tuned parameters.

Continuous-Concat MPC (CC-MPC) (Acerbo et al., 2023) uses an MLP (128, 128, 64) with ReLU activations to directly predict $\hat{z}_t$. It takes the same inputs (states and curvatures) as ours but processes them as a concatenated vector without feature conditioning. For fair comparison, the outputs are passed through the same Physics Guard and trained using only the imitation loss (7).

Continuous-Backbone MPC (CB-MPC) uses the same backbone as ours (GRU-based history encoder and FiLM-conditioned Extractor), but directly regresses the MPC parameters $\hat{z}_t$ via a linear layer without strategic bases and a gating network, followed by the same Physics Guard. The model is trained using only the imitation loss (7), isolating the benefit of structured strategic decomposition over naive end-to-end regression.