

# RL Warm-Start for Trajectory Optimization and Control: Decision Transformer + SCP vs. IQL-LSTM + MPC/SCP

February 18, 2026

## Project Description: Transformer Warm-Starts for Constrained Vehicle Trajectory Optimization

**Problem and Motivation.** This project investigates whether an offline sequence model can amortize nonlinear constrained trajectory optimization for autonomous vehicles by producing high-quality warm-start trajectories. Sequential Convex Programming (SCP) is a widely used approach for constrained trajectory generation, but it is sensitive to initialization and can require many iterations to converge in challenging nonconvex scenarios such as aggressive maneuvers and obstacle avoidance. The goal is to learn a neural trajectory prior that significantly reduces SCP runtime and failure rates while preserving hard constraint satisfaction through optimization.

**Domain and Data.** We focus on vehicle trajectory generation in simulation using a nonlinear vehicle dynamics model (e.g., a bicycle model parameterized for a 2019 VW GTI [1]). A synthetic dataset will be created by sampling randomized planning instances (initial states, goals, road geometry, obstacles, friction and actuation limits) and solving each with an SCP-based optimizer to produce feasible, near-optimal state–control trajectories. Each dataset entry will include:

- A context vector describing the planning problem,
- Full state sequence ( $x_{0:T}$ ) and control sequence ( $u_{0:T-1}$ ),
- Cost-to-go and constraint-to-go signals for conditioning.

This follows the sequence-modeling reinforcement learning framing and warm-start trajectory optimization template in the literature [2,3,4].

**Methodology.** We will train a Decision Transformer–based trajectory model that maps context and conditioning signals to full-horizon state and control sequences. At inference, this learned model will provide warm-start trajectories for SCP, which will refine them to satisfy dynamics and safety constraints [7]. Baselines will include naive initializations such as straight-line or zero control, as well as shifted prior solutions. If time permits, we will explore online finetuning inspired by Online Decision Transformer [5] to adapt to dynamic environments.

**Related Work.** Decision Transformer demonstrates that offline reinforcement learning can be reframed as sequence modeling with Transformer architectures for policies conditioned on return [2,8]. Trajectory Transformer extends this to long-horizon planning via model-based decoding [3,13]. In the control domain, transformer priors have been used to warm-start nonlinear solvers, with the Autonomous Rendezvous Transformer (ART) showing reduced iterations and improved solution quality when combined with SCP in spacecraft trajectory optimization [0,1]. Transformer MPC approaches also illustrate the benefit of learning-based warm starts in optimal control loops [15].

**Evaluation Plan.** We will quantitatively evaluate warm-start methods using:

- SCP iterations to convergence,
- Wall-clock runtime,
- Feasibility success rates,
- Final cost relative to baselines.

Qualitative results will include trajectory visualizations showing obstacle avoidance, smoothness, and constraint satisfaction. Robustness tests will assess generalization to unseen obstacles and friction variations.

## Plan of Action

1. **Vehicle Model Specification.** Implement a vehicle dynamics model (e.g., kinematic or dynamic bicycle) suitable for trajectory optimization; validate with basic simulations.
2. **SCP Solver Setup.** Adopt an open-source SCP solver such as the Implementation in `acados`, `FORCES Pro`, or a custom successive convexification implementation. Ensure support for state/control constraints.
3. **Scenario Dataset Design.** Define distributions for initial states, goals, road geometry, obstacles, and constraint parameters. Generate a large offline dataset (e.g., 5,000–20,000 solved instances) by solving each instance with the SCP solver, and record full state–control trajectories with cost/constraint signals.
4. **Transformer Architecture.** Implement a Decision Transformer model (or ART variant) that predicts full trajectory sequences conditioned on problem context, cost-to-go, and constraint-to-go signals.
5. **Training Pipeline.** Train the transformer on the synthetic dataset with supervised training aimed at minimizing state and control prediction error. Validate generalization on held-out scenarios.
6. **Warm-Start Inference Integration.** Integrate the trained model with the SCP solver. For new planning problems, generate warm-start trajectories and refine them via SCP. Benchmark against baselines.
7. **Evaluation and Analysis.** Measure solver iterations, runtime, feasibility, and final cost. Produce performance plots, comparison tables, and qualitative trajectory visualizations.
8. **Extensions (Optional).** Explore online adaptation strategies (e.g., Online Decision Transformer) if time permits to improve performance under distribution shift.

## References

1. J.K. Subosits and J.C. Gerdes, “Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers”, *IEEE Trans. Intelligent Vehicles*, 2021.
2. L. Chen et al., “Decision Transformer: Reinforcement Learning via Sequence Modeling,” NeurIPS 2021. :contentReference[oaicite:0]index=0

3. M. Janner, Q. Li, S. Levine, “Offline Reinforcement Learning as One Big Sequence Modeling Problem,” NeurIPS 2021. :contentReference[oaicite:1]index=1
4. T. Guffanti, D. Gammelli, S. D’Amico, M. Pavone, “Transformers for Trajectory Optimization with Application to Spacecraft Rendezvous,” IEEE Aerospace Conference 2024 (ART). :contentReference[oaicite:2]index=2
5. Q. Zheng, A. Zhang, A. Grover, “Online Decision Transformer,” ICML 2022. :contentReference[oaicite:3]index=3
6. D. Celestini et al., “Generalizable Spacecraft Trajectory Generation via Multimodal Learning with Transformers,” 2024. :contentReference[oaicite:4]index=4
7. Y. Mao, M. Szmuk, B. Acikmese, “Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties,” arXiv, 2016.
8. (Optional) TransformerMPC: Accelerating MPC via Transformers for warm start and constraint handling. :contentReference[oaicite:5]index=5