Noninvasive Model Adaptation via Two-Stage Optimization

November 18, 2025

Abstract

We consider a generative model adaptation problem where a pre-trained predictor remains fixed and only a thin layer of adaptation can be added. We formulate this as a two-stage stochastic optimization problem. The outer stage solves a risk-aggregated master problem that learns scene-specific weights on a set of metrics by minimizing an RU-CVaR objective over the simplex, using Benders cuts built from inner solutions. The inner stage is a planner that, given candidate weights, evaluates the corresponding weighted combination of metrics over a convex, compact feasible set and returns both objective values and subgradients. In contrast to standard fine-tuning or feature-based adaptation, which modifies predictor parameters or trains an unconstrained neural head on top of predictor outputs, our approach adapts only the "second stage decision" weights, keeping the base model unchanged. We give conditions under which this two-stage procedure is well-posed and stable and show that it can be instantiated in domains where metrics admit conic representations and whose feasibility sets are convex. A safety-focused autonomous-driving application illustrates these properties: the adapted planner reduces speed-limit and clearance violations and improves feasibility at comparable comfort, while matching or outperforming baselines that fine-tune predictor scores or learn direct trajectory policies, demonstrating the practical value of two-stage optimization for noninvasive model adaptation.

Keywords: Two-stage robust optimization; autonomous driving; scene-adaptive planning; Benders decomposition; RU–CVaR; conic optimization; linear probes; random features; hypernetworks.

1 Introduction

Motion planning must reconcile multiple objectives that vary between scenes. Fixed metric weights are insufficient because the geometry of the road, the density of traffic, and the preferences of the riders change. A two-stage stochastic program is presented that assigns scene-dependent weights through an affine map of scene features and controls tail risk through a CVaR objective. Efficiency

is achieved with an outer approximation using benders cuts, and stability is ensured by a Bayesian trust region over the affine parameters. Preference supervision is integrated during parameter estimation; deployment remains convex and data-efficient.
We formulate meta learning for planning as a ***two-stage robust optimization*** *and solve it via* ***Benders decomposition****, designing an* ***affine policy over intermediate-layer outputs*** *with an* ***offline Trajecton++-supervised prior****.*

1.1 Main Research Question

Given a driving scene, the goal is to produce a feasible trajectory together with a set of metric weights that are optimal for this scene while controlling tail risk. The hypothesis is that an affine contextual rule coupled with a convex planning core and CVaR risk control achieves this goal while preserving tractable optimization.

2 Literature Review

Meta-learning enables rapid adaptation of tasks in behavior prediction [6]. Comfort and safety surrogates built from acceleration and jerk-admitted convex formulations [1]. CVaR and the Benders/outer approximation are classical tools for risk-aware stochastic optimization. The present work integrates these elements by learning an affine mapping from scene features to weights, solving a convex recourse at the inner level, and optimizing a risk-aware convex master with linear Benders cuts.

3 Conic Atom Meta-Policy (CAMP): Unified Theory and End-to-End Training

Objects, notation, and provenance (all symbols declared). A *scene* $\xi \in \Xi$ provides map and dynamic context (lanes, drivable area, agents, signals). The decision variable is the stacked trajectory/control $y \in \mathbb{R}^n$ (e.g., $y = [q_0^\top, \dots, q_T^\top]^\top$, planar waypoints). The *feasible set* $\mathcal{Y}(\xi)$ is convex and compact and is constructed from scene-conditioned linear/second-order constraints:

$$\mathcal{Y}(\xi) = \big\{ y \;\big|\; \mathbf{E}(\xi)y = \mathbf{f}(\xi),\ \mathbf{G}(\xi)y \le \mathbf{h}(\xi),\ \|\mathbf{A}_m(\xi)y - \mathbf{b}_m(\xi)\|_2 \le u_m(\xi)\ \forall m \big\}.$$

Matrices $\mathbf{E}, \mathbf{G}, \mathbf{A}_m$ and vectors $\mathbf{f}, \mathbf{h}, \mathbf{b}_m, u_m$ are *computed once from the scene* (map geometry, safety margins, kinodynamic envelopes) and *frozen* inside each inner solve. Kinematic linear operators are the *natural-boundary* difference matrices $\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j$ (velocity/acceleration/jerk), applied per-axis with Kronecker products; windowing/filters and domain lifts are linear and absorbed into $L(\xi)$ below.

Composite atoms and their conic epigraphs. For $r = 1, \dots, R$, each atom is a convex composite

$$a_r(\xi, y) \;=\; h_r\big(T_r(\xi)y + t_r(\xi)\big), \qquad T_r(\xi) = M_r(\xi)L(\xi),$$

where $L(\xi)$ stacks scene-independent linear operators (e.g., $\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j$, windowing), and $M_r(\xi)$ selects/weights components (e.g., lane-local windows, per-agent blocks). Offsets $t_r(\xi)$ encode references (e.g., $-y_{\text{ref}}$) or limits. Each $h_r : \mathbb{R}^{p_r} \to \mathbb{R} \cup \{+\infty\}$ admits a standard conic epigraph (QP/SOCP/LP); see Appendix A. Stack $\tilde{A}(\xi, y) = [a_1(\xi, y), \dots, a_R(\xi, y)]^\top \in \mathbb{R}^R$.

Scene→weights maps and parameterization. The *atom weights* $w(\xi) \in \Delta_{R-1} := \{w \in \mathbb{R}^R_{\geq 0} \mid \mathbf{1}^\top w = 1\}$ act only as coefficients in the inner objective; thus changing w *never affects inner convexity.* Let $\phi(\xi) \in \mathbb{R}^d$ denote the features of the external predictor *frozen* (Trajectron ++), extracted in its embedding layer; optionally candidates $\{y^{(k)}\}$ are used for warm starts. We use one of the following maps (all produce $w(\xi) \in \Delta_{R-1}$):

$$\textbf{(L)}\ w(\xi) = \operatorname{softmax}\big(\Theta\, \phi(\xi)\big), \quad \Theta \in \mathbb{R}^{R \times d}, \tag{1}$$

$$\textbf{(P)}\ w(\xi) = \operatorname{softmax}\big(\Theta\, \psi(\phi(\xi))\big), \quad \psi : \mathbb{R}^d \to \mathbb{R}^D \text{ (fixed lift; polynomial/RFF)}, \tag{2}$$

$$\textbf{(H)}\ w(\xi) = \Pi_\Delta\big(h_\vartheta(\phi(\xi))\big), \quad h_\vartheta : \mathbb{R}^d \to \mathbb{R}^R \text{ (small hypernetwork)}. \tag{3}$$

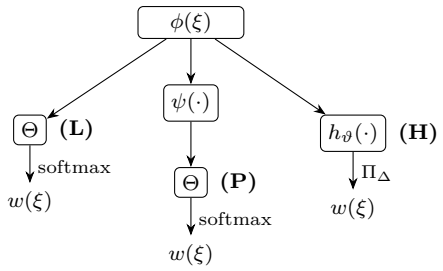

Figure 1: Head comparison: (L) linear softmax; (P) fixed nonlinear lift then linear; (H) hypernetwork with simplex projection.

Provenance and dimensions: (i) Θ (capital) is the *learned parameter matrix* (dimensions shown above). (ii) $w(\xi)$ is an R-vector (simplex). (iii) θ (lowercase, below) is a *scalar epigraph* variable in the master problem, one per scene/batch by construction.

Inner conic oracle and subgradients in w. Given (ξ, w), the inner value is the conic program

$$Q(\xi, w) = \min_{y \in \mathcal{Y}(\xi)} \langle w, \tilde{A}(\xi, y) \rangle, \qquad y(\xi; w) \in \arg\min .$$

By Danskin's theorem, the vector $g(\xi, w) := \tilde{A}(\xi, y(\xi; w)) \in \partial_w Q(\xi, w)$ is a valid *Benders* subgradient at w. A supporting cut at $\hat{w}$ is

$$\theta \;\geq\; Q(\xi, \hat{w}) + g(\xi, \hat{w})^\top \big(w - \hat{w}\big), \tag{4}$$

with *scalar* epigraph variable θ upper-bounding the scalar $Q(\cdot)$; this is why θ is 1-D.

Theorem 1 (Inner convexity invariance)**.** *If $\mathcal{Y}(\xi)$ is convex and compact and each atom $a_r(\xi, y)$ has a conic epigraph, then for any measurable $w(\xi) \in \Delta$ the inner value $Q(\xi, w) = \min_{y \in \mathcal{Y}(\xi)} \langle w, \tilde{A}(\xi, y) \rangle$ is a conic convex program in y, while $Q(\xi, w)$ is concave and polyhedral in w. The supporting inequality* (4) *with subgradient $g = \tilde{A}(\xi, y^\star)$ is valid.* (Full proof in Appendix C.1.)

Theorem 2 (Linear-softmax as conditional exponential family)**.** *If outcomes follow a multinomial exponential family with sufficient statistic $\phi(\xi)$, the MLE is $w(\xi) = \mathrm{softmax}(\Theta\phi(\xi))$. The conditional cross-entropy is convex in Θ and strictly proper (calibrated).* (Ffull proof in Appendix C.2.)

Theorem 3 (Lifted-linear universality with controlled complexity)**.** *Let $\psi : \mathbb{R}^d \to \mathbb{R}^D$ be a fixed lift (e.g., polynomial up to degree p, or random Fourier features). Consider $w(\xi) = \mathrm{softmax}(\Theta\psi(\phi(\xi)))$. Then:*

(a) ***Universality.*** *For any continuous $w^\star : \Xi \to \Delta$ on compact Ξ and any $\varepsilon > 0$, there exist D and Θ such that $\|w(\xi) - w^\star(\xi)\|_1 < \varepsilon$ for all $\xi \in \Xi$.*

(b) ***Generalization.*** *If $\|\psi\|_\infty \leq B_\psi$ and $\|\Theta\|_F \leq B$, the Rademacher complexity of the logit class is $O(BB_\psi/\sqrt{n})$, yielding a cross-entropy generalization gap of the same order.*

(c) ***Stable outer gradients.*** *For per-scene cut slope g,*

$$\nabla_\Theta(g^\top w) = \big(\mathrm{diag}(w) - ww^\top\big)\, g \;\cdot\; \psi(\phi(\xi))^\top, \qquad \|\,\mathrm{diag}(w) - ww^\top\,\|_2 \leq \tfrac{1}{4}.$$

Proof of Theorem 3. (a) For polynomial lifts, monomials form an algebra that separates points and contains constants; by Stone–Weierstrass, polynomials are dense in $C(K)$ on compact K. Approximate each target logit f_r uniformly by $\Theta_r^\top \psi(\phi(\xi))$. Softmax is Lipschitz on bounded sets and invariant to adding a constant to all logits, so uniform logit approximation implies uniform probability approximation. For RFF, random trigonometric features approximate a shift-invariant RKHS uniformly with error $O(1/\sqrt{D})$, giving the same conclusion.
(b) The class $\{\Theta\psi(\cdot) : \|\Theta\|_F \leq B, \|\psi\|_\infty \leq B_\psi\}$ has empirical Rademacher complexity $\lesssim BB_\psi/\sqrt{n}$ by linearity and contraction; proper composite losses inherit these bounds.
(c) Let $z = \Theta\psi(\phi)$ and $w = \mathrm{softmax}(z)$. The Jacobian is $J = \mathrm{diag}(w) - ww^\top$ with $\|J\|_2 \leq 1/4$. By the chain rule, $\nabla_\Theta(g^\top w) = J^\top g\; \psi(\phi)^\top$, which is linear in parameters and norm-bounded. □

Theorem 4 (Hypernetwork vs. lifted-linear)**.** *(H) is more expressive but has larger Lipschitz/covering complexity; it typically requires stronger regularization to match the stability of (P). In Benders, simplex-projection kinks can amplify cut noise near simplex faces.* (Sketch; full proof in Appendix C.3.)

Theorem 5 (Identifiability under atom collinearity)**.** *If atom vectors are nearly collinear across $\mathcal{Y}(\xi)$ (ill-conditioned atom Gram), weights are weakly identifiable. Entropy/Dirichlet regularization on logits and ridge (Tikhonov) on Θ improve conditioning and identifiability.* (Sketch; full proof in Appendix C.4.)

Theorem 6 (Blocks / product-of-simplices)**.** *For $W = [w^{(1)}, \dots, w^{(S)}] \in \Delta^S$, the subgradient stacks as $g = [\tilde{A}^{(1)}; \dots; \tilde{A}^{(S)}]$, and outer gradients remain linear-in-parameters for (L-b)/(P-b).* (Sketch; full proof in Appendix C.5.)

Outer risk-robust master and end-to-end learning. For scenes $\{\xi_s\}_{s=1}^S$, we minimize a RU–CVaR objective with standard epigraph ($t_s, u_s \geq 0$) and Benders cuts (4):

$$\min_{\text{map params}, \{t_s, u_s, \theta_s\}} \frac{1}{S} \sum_{s=1}^{S} \left[t_s + \tfrac{1}{1-\alpha}\, u_s \right] \quad \text{s.t.} \quad \theta_s \geq Q(\xi_s, \hat{w}_s) + g_s^\top (w_s - \hat{w}_s),\ u_s \geq \theta_s - t_s,\ w_s = \textsc{Map}(\phi(\xi_s)).$$

End-to-end: with logits $z_s = \Theta\, \phi_s$ (or $\Theta\, \psi(\phi_s)$) and $w_s = \mathrm{softmax}(z_s)$, each cut term $g_s^\top w_s$ has gradient

$$\nabla_\Theta (g_s^\top w_s) = \big(\mathrm{diag}(w_s) - w_s w_s^\top\big) g_s \ \cdot\ (\mathrm{feature}_s)^\top,$$

preserving linear-in-parameters structure (stable, regularizable). *Convex distillation:* alternatively solve for per-scene w_s in the convex master, then fit Θ via $\sum_s \mathrm{KL}(w_s \,\|\, \mathrm{softmax}(\Theta\, \mathrm{feature}_s))$.

Block/product-of-simplices (when a single w is insufficient). If weights must vary across S segments (e.g., time windows/lanes/agents), partition atoms into blocks $\tilde{A}^{(s)}(\xi, y) \in \mathbb{R}^R$ and define

$$Q(\xi, W) = \min_{y \in \mathcal{Y}(\xi)} \sum_{s=1}^{S} \langle w^{(s)}, \tilde{A}^{(s)}(\xi, y) \rangle, \quad W = [w^{(1)}; \dots; w^{(S)}] \in \Delta_{R-1}^S.$$

Maps generalize as $w^{(s)} = \mathrm{softmax}(\Theta^{(s)} \phi)$ or via a hypernetwork with blockwise simplex projection. The subgradient stacks as $g = [\tilde{A}^{(1)}(\xi, y^*); \dots; \tilde{A}^{(S)}(\xi, y^*)]$, and the scalar epigraph θ and cut (4) remain unchanged (only the cut's dimension grows).

Domain universality (sufficient conditions). CAMP applies whenever: (C1) $\mathcal{Y}(\xi)$ is convex/compact; (C2) each h_r has a standard conic epigraph; (C3) the scene-to-weight map outputs in Δ (or a product of simplices); (C4) metrics

and feasible sets depend continuously on ξ. To instantiate a new domain, provide the tuple

$$\big(y, \{L_b\}, M_r, t_r, h_r, \mathcal{Y}(\xi)\big), \quad T_r(\xi) = M_r(\xi)L(\xi),$$

choose h_r from Appendix A, and list feasibility blocks of $\mathcal{Y}(\xi)$. Concrete domains with *variable meanings and formulas* are detailed in Appendix B.

4 Problem setup and data

4.1 Scene, variables, and feasible set (compact)

Scene and features. A scene $\xi \in \Xi$ provides map layers and dynamic context (lanes, boundaries, posted speed, nearby agents, etc.). Fixed features $\phi(\xi) \in \mathbb{R}^d$ are computed once per scene and remain frozen during inner solves.

Trajectory stacking and kinematics. Let $y = [q_0^\top, \dots, q_T^\top]^\top \in \mathbb{R}^{2(T+1)}$ stack planar positions. Finite-difference operators (with natural boundary conditions) map positions to velocity, acceleration, and jerk:

$$\boldsymbol{v} = \mathbf{D}_v y, \qquad \boldsymbol{a} = \mathbf{D}_a y, \qquad \boldsymbol{j} = \mathbf{D}_j y.$$

The explicit banded forms of $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$ are deferred to Appendix D.

Scene-induced feasible set. All linear/SOC operators are computed from ξ and frozen within each inner solve:

$$\mathcal{Y}(\xi) = \Big\{ y \mid \mathbf{E}(\xi)y = \mathbf{f}(\xi),\ \mathbf{G}(\xi)y \le \mathbf{h}(\xi),\ \big\|\mathbf{A}_m(\xi)y - \mathbf{b}_m(\xi)\big\|_2 \le u_m(\xi)\ \ \forall m \Big\}. \tag{5}$$

Typical elements include boundary / initialization constraints, speed / acceleration envelopes, lanekeeping corridors, and SOC distance margins; explicit block structures appear in the appendix E.

Normalized convex metrics (pointer). We use normalized convex metrics (and their atomic expansion) introduced later in5.1; all terms are convex and QP/SOCP-representable in y under (5).

Standing assumptions.

(A1) For each ξ, $\mathcal{Y}(\xi)$ in (5) is not empty, convex, and compact.

(A2) Each metric/atom is convex and lower semicontinuous on $\mathcal{Y}(\xi)$; per-scene operators are uniformly bounded.

(A3) (Optional) A small Tikhonov term $\frac{\varepsilon}{2}\|y\|_2^2$ may be added to enforce strong convexity/uniqueness.

4.2 Operators, scaling, and normalization (compact)

Time grid and stacking. We use a uniform grid with step $\Delta t > 0$ and stack $y \in \mathbb{R}^{2(T+1)}$ as above.

Difference operators with natural boundary. Velocities, accelerations, and jerk follow $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$; 1D stencils and their 2×2 block assembly are listed in Appendix D.

SOC epigraphs and hinges. Second-order cone epigraphs $s \geq \|Ly - b\|_2$ and hinges $s \geq \|Ly\|_2 - \tau,\ s \geq 0$ are used for speed / acceleration envelopes and distance margins; their scene-dependent block structures are given in the appendix E.

Discrete integration and RMS. For any time series z_t: $\Delta t \sum_{t=0}^{T-1} \|z_t\|_2^2 = \|\sqrt{\Delta t}\, z\|_2^2$. For a window W, the RMS is $\mathrm{RMS}(z; W) = \frac{1}{\sqrt{|W|}} \|\sqrt{\Delta t}\, S_W z\|_2$, where S_W is a linear averaging operator (diagonal block).

Per-metric/atom normalization. Each convex metric/atom $a_r(\xi, y)$ is normalized by a positive scale s_r (robust median/percentile on training scenes), $\tilde{a}_r = a_r / s_r$. We stack normalized entries as $\tilde{A}(\xi, y)$ (cf.5.1) and aggregate with simplex weights; this preserves convexity and physical units.

4.3 Datasets and preprocessing

4.3.1 nuScenes

We adopt the official nuScenes splits and map layers. Each scene provides ego poses, lane geometries, boundaries, and posted speed. Our pipeline produces per scene features $\phi(\xi)$ and supervision (ground truth trajectories) as follows.

Map and geometry. We rasterize the centerlines and boundaries of the lane in the ego-centric frame, project the posted speed to a per-step limit $v_{\max}(t, \xi)$, and derive corridor bands for lanekeeping constraints (linear inequalities in (5)). The distance margins signed on the static map polygons are encoded using SOC epigraphs (Appendix E).

Dynamic context. Nearby agents are filtered by range and field-of-view; optional safety buffers are added as SOC distance margins if used in the scene. Agent statistics (counts, relative headings, min distance) are included in $\phi(\xi)$.

Time grid and horizon. We resample the trajectories to a fixed step Δt and horizon T (the values used in the experiments are specified in 5). The difference operators $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$ and the averaging windows S_W reuse this grid (Appendix. D).

Ground truth and candidates. The ground truth y_i^{gt} is extracted from future ego poses on the same grid. External candidates $\{y_i^{(k)}\}$ used for the Pred-Top1 scoring and offline policy are provided by *Trajecton++* (details and usage in 5). All optimized outputs are rescored by our convex metrics for fairness.

Feature vector $\phi(\xi)$. Unless stated otherwise, $\phi(\xi)$ concatenates the map descriptors (lane curvature, distance to centerline, post speed summary), the features of the layout of the scene (corridor width, number of lanes) and the dynamic statistics (agent counts, min distance, relative motion histograms). Features are standardized per dimension on the training split.

Quality checks. We drop scenes with incomplete map layers or missing future poses and verify that all per-scene operators used in (5) are well-formed (bounded operator norms), ensuring assumptions A1–A3.

5 Intermediate-layer policy and robust meta-learning

Highlight. We cast meta-learning as *two-stage robust optimization* and solve it by *Benders decomposition*, designing an *affine policy over intermediate-layer outputs*.See Fig. 2 for an overview.

5.1 Intermediate outputs: atomic expansion of the base metrics

We refine the four base metrics into a compact convex bank *atoms* that serves as an intermediate representation. Let $\mathcal{W} = \{W_k\}_{k=1}^{K_W}$ be temporal windows with averaging operators S_{W_k} (block diagonal), and let $\tau = \{\tau_\ell\}_{\ell=1}^{K_\tau}$ be speed-cap margins. Using the difference operators $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$ from4.1 (explicit stencils in Appx. D), define

$$a_k^{\mathrm{jerk}}(\xi, y) := \left\|\sqrt{\Delta t}\, S_{W_k}\, \mathbf{D}_j y\right\|_2^2, \qquad k = 1, \dots, K_W, \tag{6}$$

$$a_k^{\mathrm{acc}}(\xi, y) := \left\|\sqrt{\Delta t}\, S_{W_k}\, \mathbf{D}_a y\right\|_2^2, \qquad k = 1, \dots, K_W, \tag{7}$$

$$a_k^{\mathrm{rms}}(\xi, y) := \frac{1}{\sqrt{|W_k|}} \left\|\sqrt{\Delta t}\, S_{W_k}\, \mathbf{D}_a y\right\|_2, \qquad k = 1, \dots, K_W, \tag{8}$$

$$a_\ell^{\mathrm{spd}}(\xi, y) := \Delta t \sum_{t=0}^{T-1} s_{t,\ell}^2, \qquad \text{s.t. } s_{t,\ell} \geq \left\|(\mathbf{D}_v y)[t]\right\|_2 - \big(v_{\max}(t, \xi) - \tau_\ell\big),\ s_{t,\ell} \geq 0. \tag{9}$$

All $a(\xi, \cdot)$ are norms/squared norms of linear images or SOC hinges; therefore, they are convex and QP / SOCP representative under the feasible set in4.1 (block

structures in Appx. E). Stack $R = 3K_W + K_\tau$ atoms as $A(\xi, y) = [a_1, \ldots, a_R]^\top$ and normalize via per-atom scales $s_r > 0$:

$$\tilde{A}_r = a_r / s_r, \qquad \tilde{A}(\xi, y) = [\tilde{A}_1, \ldots, \tilde{A}_R]^\top.$$

Choosing W_{all} and τ_1=0 recovers the four base metrics up to scaling; thus the atomic form strictly refines the original metrics without changing the solvers.

5.2 Offline policy from Trajecton++ (supervision on intermediate outputs)

We estimate $w_{\text{off}} \in \Delta_{R-1}$ via a Bradley–Terry preference model on intermediate outputs:

$$\min_{w \in \Delta_{R-1}} \sum_{i,k} w_{i,k} \ \log\big(1 + \exp\big(w^\top(\tilde{A}(\xi_i, y_i^{\text{gt}}) - \tilde{A}(\xi_i, y_i^{(k)}))\big)\big) \ + \ \beta\|w\|_1, \quad (10)$$

with candidate weights $w_{i,k} \propto \exp(-\tau \cdot \text{ADE}(y_i^{(k)}))$ and $\beta \in [10^{-3}, 10^{-2}]$. We found pairwise fitting more robust to varying candidate sets and missing strong proposals in nuScenes with Trajecton++.

5.3 Online affine policy over intermediate outputs and RU–CVaR master

At meta-time we adapt per-scene weights via an affine policy.

$$w_i \ = \ \Theta_w \, \phi(\xi_i), \qquad w_i \in \Delta_{R-1}. \quad (11)$$

The scene-wise inner value (recourse) is

$$Q(\xi_i, w_i) \ = \ \min_{y \in \mathcal{Y}_{\xi_i}} \ \langle w_i, \tilde{A}(\xi_i, y)\rangle \ \big(+ \tfrac{\varepsilon}{2}\|y\|_2^2 \text{ if enabled one best solution}\big), \quad (12)$$

which preserves convexity and the QP/SOCP structure.

Empirical RU–CVaR master (self-contained). Given scenes $\{\xi_i\}_{i=1}^M$, the two-stage robust meta-objective reads

$$\begin{aligned} \min_{\Theta_w, \ \{w_i\}, \ \{\theta_i\}, \ \eta, \ \{s_i \geq 0\}} \quad & \eta + \frac{1}{(1-\alpha)M} \sum_{i=1}^M s_i \\ \text{s.t.} \quad & s_i \ \geq \ \theta_i - \eta, \quad i = 1, \ldots, M, \\ & \theta_i \ \geq \ Q(\xi_i, w_i), \quad i = 1, \ldots, M, \\ & w_i \ = \ \Theta_w \, \phi(\xi_i), \quad w_i \in \Delta_{R-1}. \end{aligned} \quad (13)$$

To stabilize the updates of Θ_w, we use an ellipsoidal trust region.

$$\big\|\Sigma_{\Theta_w}^{1/2}\big(\text{vec}(\Theta_w) - \mu_{\Theta_w}\big)\big\|_2 \ \leq \ \rho_{\Theta_w}, \quad (14)$$

and we may anchor w_i around the offline estimate w_{off} via either an SOC trust region $\| \Sigma_w^{1/2}(w_i - w_{\text{off}})\|_2 \le \rho_w$ or a penalty $\gamma_{\text{off}} \sum_i \|w_i - w_{\text{off}}\|_2^2$.
Unless stated otherwise, we set $\rho_{\Theta_w} = 1.0$, $\rho_w = 0.5$, and $\gamma_{\text{off}} = 0.1$. This risk aggregation follows the CVaR framework [8, 9].

5.4 Benders in machine learning: cuts, pools, and geometry

By Danskin, for any minimizer $y_i^\star \in \arg\min_{y \in \mathcal{Y}_{\xi_i}} \langle w_i, \tilde{A}(\xi_i, y)\rangle$, the slope $g_i := \tilde{A}(\xi_i, y_i^\star) \in \partial_w Q(\xi_i, w_i)$ yields the global support inequality.

$$\theta_i \;\ge\; Q_i \;+\; g_i^\top \big(w_i - \hat{w}_i\big), \qquad Q_i = \langle \hat{w}_i, g_i \rangle. \tag{15}$$

We maintain two pools: (i) a cut pool $\mathcal{K}_i$ storing $(Q_i^{(k)}, g_i^{(k)}, \hat{w}_i^{(k)})$, and (ii) an active set pool to warm start inner solves. A cut is kept if it is violated in the current iterate by more than a tolerance; stale cuts are recycled. All master constraints remain linear/SOC, so the master is convex.

Lemma (Danskin + supporting cuts). Let $Q(\xi, w) = \min_{y \in \mathcal{Y}(\xi)} \langle w, \tilde{A}(\xi, y)\rangle$ with a unique minimizer. Then any $g = \tilde{A}(\xi, y^\star) \in \partial_w Q(\xi, w)$ produces a global supporting cut $\theta \ge Q(\xi, \hat{w}) + g^\top (w - \hat{w})$. *Proof sketch* in Appx. I. See also classical references on outer approximation and subgradients [2–4, 7, 8].

5.5 Algorithms (concise)

Algorithm 1 OfflinePolicy on intermediate outputs (Trajecton++)

Inputs: scenes $\{\xi_i\}$; GT y_i^{gt}; Trajecton++ candidates $\{y_i^{(k)}, p_i^{(k)}\}_{k=1}^K$.
for each (i, k) **do**
  Compute $\tilde{A}(\xi_i, y_i^{\text{gt}})$ and $\tilde{A}(\xi_i, y_i^{(k)})$; set $w_{i,k} \propto p_i^{(k)}$.
end for
Solve (10) for $w_{\text{off}} \in \Delta_{R-1}$.

Algorithm 2 Level-A master with intermediate policy (Benders)

Variables: $\Theta_w, \{w_i\}, \{\theta_i\}, \eta, \{s_i\}$; pools $\{\mathcal{K}_i\}$ and active-set memories.
repeat
  for each scene i **do**
    Inner solve $y_i^{(*)} \in \arg\min_{y \in \mathcal{Y}_{\xi_i}} \langle w_i, \tilde{A}(\xi_i, y)\rangle$ (warm-start).
    Add $(Q_i^{(*)}, g_i^{(*)})$ with $g_i^{(*)} = \tilde{A}(\xi_i, y_i^{(*)})$ into $\mathcal{K}_i$.
  end for
  Solve the master (13) with cuts (15) (plus anchors/trust regions (14)) to update $(\Theta_w, \{w_i\}, \{\theta_i\}, \eta, \{s_i\})$.
until stopping

Instantiating the "Bayesian" trust region. We set the center at the offline optimum, $\mu_{\Theta_w} \leftarrow \mathrm{vec}(\widehat{\Theta}_w)$, and the precision to the (empirical) Fisher information of the offline preference fit at w_{off} (with damping):

$$\Sigma_{\Theta_w}^{-1} \;\propto\; \underbrace{\nabla^2 \mathcal{L}_{\mathrm{off}}(\widehat{w}_{\mathrm{off}})}_{\text{empirical Fisher / Hessian}} \;+\; \lambda_{\mathrm{damp}}\, \mathbf{I},$$

where $\mathcal{L}_{\mathrm{off}}$ is the convex offline loss in (10). Concretely, for the Bradley–Terry model in pairwise (10) it matches the logistic regression Fisher matrix. This Laplace approximation justifies the term "Bayesian trust region" while keeping the master convex and SOCP-representable. Unless stated otherwise, we use $\lambda_{\mathrm{damp}} \in [10^{-3}, 10^{-2}]$ and $\rho_{\Theta_w} = 1.0$ by default.

5.6 Diagram

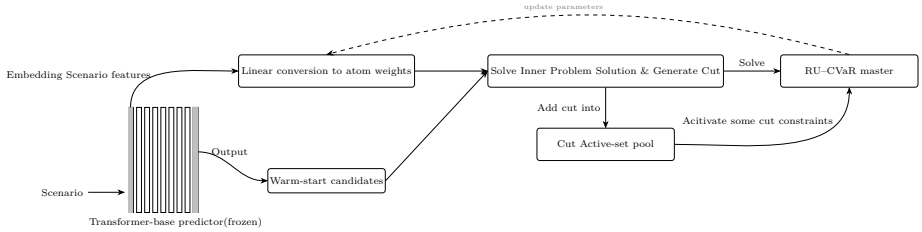

Figure 2: Architecture with fewer notations and intuitive illustrations. Original labels kept; formulas removed. The highlighted vertical slat indicates the embedding layer; its output (Embedding features $\phi(\xi)$) feeds the Scene→weights map, while Output candidates $\{y^{(k)}\}$ feed Warm-start candidates. Full training flow through Inner conic, Intermediate, Cut / Active-set pool, and RU–CVaR master (Benders) with dashed feedback is preserved.

6 Experiments

6.1 Datasets and protocol

We evaluate on **nuScenes** (official splits; see §4.3.1). All scenes are preprocessed as in §4.3: ego-centric rasterization, posted-speed projection to $v_{\max}(t, \xi)$, standardized features $\phi(\xi)$, and a fixed grid $(\Delta t, T)$ shared by the difference operators $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$ (§4.1). For each scene, *Trajecton++* provides K candidate trajectories used for Pred-Top1 scoring and as warm starts in planning methods. Unless stated otherwise, we use the defaults in Table 1 in all experiments.

6.2 External predictor: Trajecton++

We use **Trajecton++** as the external motion predictor throughout. For Pred-Top1, we directly evaluate its Top-1 output y_i^{pred}. For planning methods (Plan-*),

the Trajecton++ candidates serve only as *warm starts*; the final outputs are produced by our inner convex solver and rescored by the same convex metrics for fairness. We use **Trajecton++** as the external motion predictor throughout [5].

6.3 Methods compared

- **Pred-Top1 (Trajecton++)**: scoring-only baseline; see Alg. 3.
- **Plan-Static**: inner convex planning with a fixed global weight w (either hand tuned or w_{off} from5.2).
- **Plan-Adaptive (λ-only)**: original Level A with $\lambda = \Theta\phi(\xi)$ (no intermediate layer expansion).
- **Plan-Adaptive (intermediate)**: our method with atoms $\tilde{A}$ (5.1), affine policy $w_i = \Theta_w\phi(\xi_i)$ (§5.3), RU–CVaR master (13), and Benders cuts (15).

All planning methods use identical linearizations, map layers, and solver settings; Trajecton++ outputs are used as warm starts where applicable.

6.4 Evaluation metrics

We report ADE/FDE, feasibility rate against $\mathcal{Y}(\xi)$ (§4.1), and normalized convex metrics of the atomic families (jerk/accel/RMS/speed-cap; §5.1). All optimized outputs are rescored by the convex metrics for fairness.

6.5 Pred-Top1 (Trajecton++)

Algorithm 3 Pred-Top1 (Trajecton++ scoring only)

1: Inputs: scenes $\{\xi_i\}$; **Trajecton++** predictor providing Top-1 trajectory y_i^{pred}.
2: **for** each scene i **do**
3: Score y_i^{pred} by ADE, FDE and convex metrics $c_j(\xi_i, y_i^{\text{pred}})$; record feasibility relative to $\mathcal{Y}(\xi_i)$.
4: **end for**

6.6 Ablations

- **Offline policy**: Plan-Static with random / complex center vs. w_{off} (learned through 5.2).
- **Pools**: Benders only vs. Benders + Cut / Active-Set pools (5.4).
- **Risk objective**: ERM (mean) vs. RU–CVaR with $\alpha \in \{0.8, 0.9, 0.95\}$ in (13).

6.7 Settings and hyperparameters

Table 1 lists grid/horizon $(\Delta t, T)$, atom configuration $(K_W,\ K_\tau)$, solver tolerances, and RU–CVaR parameters. We project w_i and $\Theta_w \phi(\xi_i)$ onto the simplex at every master step using Alg. 4.

Table 1: Default settings (nuScenes).

Component	Symbol	Default
Time step / horizon	$(\Delta t, T)$	$(0.5\,\mathrm{s},\ 40)$
Windows (short/med/full)	K_W	3
Speed-cap margins	K_τ	2 ($\{0,\ 1.5\}\,\mathrm{m/s}$)
RU–CVaR level	α	0.90
Trust regions	$(\rho_{\Theta_w}, \rho_w)$	$(1.0,\ 0.5)$
Offline damping	λ_{damp}	10^{-3}
Solver tolerances		10^{-6} (primal/dual)

6.8 Main results on nuScenes

Table 2: Runtime and iterations (averaged per scene; placeholders).

Method	Inner iters / time (ms)	Master iters	Avg. cuts kept
Plan-Static (hand-tuned)			
Plan-Static (w_{off})			
Plan-Adaptive (λ-only)			
Plan-Adaptive (intermediate)			

Table 3: Main results on nuScenes (fill with numbers). Lower is better for losses; higher for feasibility.

Method	ADE ↓	FDE ↓	Feas. ↑	Convex metrics ↓
Pred-Top1 (Trajecton++)				
Plan-Static (hand-tuned)				
Plan-Static (w_{off})				
Plan-Adaptive (λ-only)				
Plan-Adaptive (intermediate)				

7 Conclusion

A convex, risk-aware planner with an affine contextual rule for weights and a Bayesian trust region has been developed. The inner convexity and linear

bender cuts permit frequent updates without breaking the correctness. A five-method evaluation suite is defined to isolate the contributions of scene features and adaptation to test time. The formulation is compatible with nuScenes; regulatory metrics are supplied in convex form for datasets with the required annotations.

References

[1] I. Bae, J. Moon, and J. Seo. Toward a comfortable driving experience for a self–driving shuttle bus. *Electronics*, 8(9):943, 2019. doi: 10.3390/electronics8090943.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] R. E. Danskin. The theory of max-min, with applications. In *Programming Under Uncertainty and Applications*, pages 10–52. Springer, 1967.

[4] A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[5] B. Ivanovic, J. Harrison, and M. Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning, 2023. URL `https://arxiv.org/abs/2209.11820`.

[6] B. Ivanovic, J. Harrison, and M. Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta–learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023. URL `https://arxiv.org/abs/2209.11820`.

[7] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[8] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

[9] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

Table 4: Solver-ready epigraphs for $h_r(u)$ (plug into $a_r(\xi, y) = h_r(T_r y + t_r)$).

$h_r(u)$	Epigraph / auxiliaries	Cone
Quadratic norm $\|Su\|_2^2$	QP (or $t \geq \|Su\|_2^2$)	QP
Euclidean norm $\|Su\|_2$	$t \geq \|Su\|_2$	SOCP
Hinge on norm $\max\{0, \|u\|_2 - \tau\}$	$s \geq \|u\|_2 - \tau;\ s \geq 0;\ t \geq s$	SOCP+lin
Polyhedral max $\max_j(a_j^\top u + b_j)$	$t \geq a_j^\top u + b_j\ \forall j$	LP
Distance to set $\text{dist}(u, \mathcal{S})$	$\exists x \in \mathcal{S}:\ \|u - x\|_2 \leq s;\ t \geq s$	SOCP(+set)
$\text{CVaR}_\alpha(u)$	$t + \frac{1}{(1-\alpha)S} \sum_s z_s,\ z_s \geq u_s - t,\ z_s \geq 0$	LP/SOCP

A Conic epigraph recipes (atoms h_r)

B Cross-domain instantiation with variable meanings

Table 5: D1–D3: variable meanings, atoms, and feasibility.

Domain	Atoms $a_r(\xi, y) = h_r(T_r y + t_r)$ (with variable meanings)	Cone	Feasible set $\mathcal{Y}(\xi)$ (examples)
D1 Industrial manipulation/control	*Decision* $y = [q_0^\top, \ldots, q_T^\top]$ (joint or EE positions). Jerk power: $T = \text{blkdiag}(S_W)\mathbf{D}_j$, $t = 0$, $h(u) = \|u\|_2^2$. Accel power: $T = \text{blkdiag}(S_W)\mathbf{D}_a$, $t = 0$, $h(u) = \|u\|_2^2$. RMS tracking: $T = \text{blkdiag}(S_W)$, $t = -y_{\text{ref}}$, $h(u) = \|u\|_2$. Speed-limit: $T = C_v\mathbf{D}_v$, $t = 0$, $h(u) = \max\{0, \|u\|_2 - \bar{v}\}$.	QP/SOCP	Joint/EE boxes; corridor polytope; boundary knots (lin/SOC).
D2 Warehouse AGV (path+schedule)	*Decision* $y = [x_{0:T}, y_{0:T}]$ or speeds $v_{0:T}$. Energy: $T = \text{blkdiag}(I, \mathbf{D}_a)$, $t = 0$, $h(u) = \|W^{1/2}u\|_2^2$. Comfort: $T = S_W\mathbf{D}_a$, $t = 0$, $h(u) = \|u\|_2$. Schedule slack: $T = \mathbf{1}^\top C y$, $t = 0$, $h(u) = \max\{0, \tau_{\min} - u\}$.	QP/SOCP/LP	Kinematic envelopes; aisle polygons; headway windows (lin).
D3 Power dispatch (OPF)	*Decision* $y = p$ (generator outputs). Gen cost: $T = I$, $t = 0$, $h(u) = c^\top u + \frac{1}{2}u^\top H_c u$. Line flow limit: $T = H$ (DC/AC sensitivities), $t = 0$, $h(u) = \max\{0, \|u\| - \bar{f}\}$. Renewable shortfall-CVaR: $T = [d_\omega - \mathbf{1}^\top(\cdot) - r_\omega]_\omega$, $t = 0$, $h = \text{CVaR}_\alpha$.	QP/LP/SOCP	Power balance; ramp/box; (relaxed) AC envelopes (SOC).

Table 6: D4–D6: variable meanings, atoms, and feasibility.

Domain	Atoms and meanings	Cone	Feasible set $\mathcal{Y}(\xi)$
D4 Portfolio (asset weights)	*Decision* $y = x \in \Delta$ (asset weights). Mean–variance: $T = I$, $t = 0$, $h(u) = \frac{1}{2}u^\top \Sigma u - \mu^\top u$. Loss-CVaR: $T = [-r_s^\top]_s$, $t = 0$, $h = \mathrm{CVaR}_\alpha$.	QP/LP	Budget/box/turnover limits (lin).
D5 Radiotherapy (IMRT)	*Decision* y (beamlet fluence). Dose: $d = Dy$. PTV underdose: $T = -D$, $t = d^\star$, $h(u) = \sum_j \max\{0, u_j\}$. OAR overdose: $T = D$, $t = -\bar{d}$, $h(u) = \sum_j \max\{0, u_j\}$. Smoothness: $T = G$, $t = 0$, $h(u) = \|u\|_2^2$.	LP/QP	Fluence nonnegativity; machine bounds; DVH-type lin.
D6 Renewables / storage	*Decision* $y = [u^+; u^-]$ (charge/discharge power). Ramp/energy: $T = \mathrm{blkdiag}(I, \mathbf{D})$, $t = 0$, $h(u) = \|W^{1/2}u\|_2^2$. Net shortfall-CVaR: $T = [d_\omega - g(\cdot)]_\omega$, $t = 0$, $h = \mathrm{CVaR}_\alpha$.	QP/LP/SOCP	SOC dynamics; efficiency; box and rate (lin/SOC).

C Proofs of Theorem

C.1 Proof of Theorem 1

Proof. Fix ξ. Since each $a_r(\xi, y)$ has a conic epigraph, $\sum_r w_r a_r(\xi, y)$ is convex in y for any $w \in \Delta$, and with $\mathcal{Y}(\xi)$ convex/compact, the minimum exists. Define $\mathcal{S}(\xi) = \{\tilde{A}(\xi, y) : y \in \mathcal{Y}(\xi)\}$, convex/compact by continuity. Then

$$Q(\xi, w) = \min_{y \in \mathcal{Y}(\xi)} \langle w, \tilde{A}(\xi, y)\rangle = \min_{z \in \mathcal{S}(\xi)} \langle w, z\rangle$$

is the support function of $\mathcal{S}(\xi)$ at w, hence concave and piecewise-linear in w. By Danskin, if $y^\star \in \arg\min_{y \in \mathcal{Y}(\xi)} \langle \hat{w}, \tilde{A}(\xi, y)\rangle$, then $g = \tilde{A}(\xi, y^\star) \in \partial_w Q(\xi, \hat{w})$ and $Q(\xi, w) \le Q(\xi, \hat{w}) + g^\top(w - \hat{w})$, which is exactly (4). □

C.2 Proof of Theorem 2

Proof. Let the natural parameter be $z(\xi) = \Theta\phi(\xi)$ with log-partition $A(z) = \log \sum_k e^{z_k}$. The negative log-likelihood equals the conditional cross-entropy and is convex in Θ by convexity of A. First-order conditions give the GLM moment-matching equation, whose solution has $w(\xi) = \mathrm{softmax}(z(\xi))$. Strict propriety (calibration) follows from proper composite loss theory for the multinomial case. □

C.3 Proof of Theorem 4

Proof. Let $h_\vartheta : \mathbb{R}^d \to \mathbb{R}^R$ be an L-layer ReLU network with weights W_ℓ. ReLU is 1-Lipschitz and positively homogeneous, so $\|h_\vartheta(u) - h_\vartheta(v)\|_2 \le \big(\prod_{\ell=1}^L \|W_\ell\|_2\big)\|u - v\|_2$. Euclidean projection onto a closed convex set is nonexpansive; the simplex

projection Π_Δ is 1-Lipschitz. Thus $\|w(u) - w(v)\|_2 \le \big(\prod_{\ell=1}^{L} \|W_\ell\|_2\big)\|u - v\|_2$. Spectral-complexity generalization bounds for ReLU nets scale with $\prod_\ell \|W_\ell\|_2$ (and depth), typically larger than the linear-in-parameter (P) head, hence stronger regularization is required. Since Π_Δ is piecewise affine with kinks at simplex faces, gradients can change abruptly near boundaries, which can amplify cut noise in Benders updates. □

C.4 Proof of Theorem 5

Proof. Write $Q(\xi, w) = \min_{y \in \mathcal{Y}(\xi)} \langle w, \tilde{A}(\xi, y)\rangle$ and let $\mathcal{S}(\xi) = \{\tilde{A}(\xi, y) : y \in \mathcal{Y}(\xi)\}$. Consider the Gram matrix $G = \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top]$ for $\boldsymbol{z} \sim \mathcal{S}(\xi)$. If $\lambda_{\min}(G)$ is small (atoms nearly collinear), there exist $w \neq w'$ with $\|w - w'\|_2 = 1$ such that $|(w - w')^\top \boldsymbol{z}| \le \sqrt{\varepsilon}\,\|\boldsymbol{z}\|_2$ for all $\boldsymbol{z}$ in the support, implying $|Q(\xi, w) - Q(\xi, w')| \lesssim \sqrt{\varepsilon}\,\mathbb{E}\|\boldsymbol{z}\|_2$. Hence multiple weights are observationally indistinguishable (weak identifiability). Entropy/Dirichlet on logits and ridge on Θ increase curvature/Fisher information, improving conditioning. □

C.5 Proof of Theorem 6

Proof. Let $W = [w^{(1)}, \dots, w^{(S)}]$ and define $Q(\xi, W) = \min_{y \in \mathcal{Y}(\xi)} \sum_{s=1}^{S} \langle w^{(s)}, \tilde{A}^{(s)}(\xi, y)\rangle$. Convexity in y follows from conic representability of atoms; concavity and polyhedrality in W follow since Q is the support function of $\{[\tilde{A}^{(1)}(\xi, y); \dots; \tilde{A}^{(S)}(\xi, y)] : y \in \mathcal{Y}(\xi)\}$. Any minimizer $y^\star$ yields a stacked subgradient $g = [\tilde{A}^{(1)}(\xi, y^\star); \dots; \tilde{A}^{(S)}(\xi, y^\star)]$ and the stated cut. For (L-b)/(P-b) with logits $z^{(s)} = \Theta^{(s)}\psi^{(s)}(\phi(\xi))$,

$$\nabla_{\Theta^{(s)}}(g^\top W) = \big(\mathrm{diag}(w^{(s)}) - w^{(s)}{w^{(s)}}^\top\big)\, g^{(s)} \left(\psi^{(s)}(\phi(\xi))\right)^\top,$$

so the outer update remains linear in parameters. □

D Difference operators with natural boundary

We give 1D stencils on a uniform grid with step Δt, then we assemble 2D blocks by duplication. Let $q = [q_0, \dots, q_T]^\top \in \mathbb{R}^{T+1}$, and I_2 be the identity 2×2.

Velocity (first difference). $\mathbf{D}^{(1)} \in \mathbb{R}^{T \times (T+1)}$ with rows

$$(\mathbf{D}^{(1)} q)_t \;=\; \frac{q_{t+1} - q_t}{\Delta t}, \quad t = 0, \dots, T-1,$$

i.e., $-1/\Delta t$ on the main diagonal and $+1/\Delta t$ on the superdiagonal. Define $\mathbf{D}_v := \mathrm{blkdiag}(\mathbf{D}^{(1)}, \mathbf{D}^{(1)}) \in \mathbb{R}^{2T \times 2(T+1)}$.

Acceleration (second difference). $\mathbf{D}^{(2)} \in \mathbb{R}^{(T-1) \times (T+1)}$ with rows

$$(\mathbf{D}^{(2)} q)_t \;=\; \frac{q_{t+1} - 2q_t + q_{t-1}}{\Delta t^2}, \quad t = 1, \dots, T-1,$$

i.e., $[1, -2, 1]/\Delta t^2$ centered on t, giving $\mathbf{D}_a := \mathrm{blkdiag}(\mathbf{D}^{(2)}, \mathbf{D}^{(2)})$.

Jerk (third difference). $\mathbf{D}^{(3)} \in \mathbb{R}^{(T-2)\times(T+1)}$ with rows

$$(\mathbf{D}^{(3)}q)_t \;=\; \frac{q_{t+2} - 3q_{t+1} + 3q_t - q_{t-1}}{\Delta t^3}, \quad t = 1, \dots, T-2,$$

i.e., $[-1, 3, -3, 1]/\Delta t^3$ centered on t, and $\mathbf{D}_j := \mathrm{blkdiag}(\mathbf{D}^{(3)}, \mathbf{D}^{(3)})$.

Natural boundary. Ghost states are not introduced; boundary rows that would reference missing indices are simply not present, yielding sizes.

$$\mathbf{D}_v : 2T \times 2(T{+}1), \quad \mathbf{D}_a : 2(T{-}1) \times 2(T{+}1), \quad \mathbf{D}_j : 2(T{-}2) \times 2(T{+}1).$$

E Scene-induced constraint blocks and SOC epigraphs

Generic SOC epigraph. For any linear map L and vector b, the epigraph $\{(y, s) \mid s \geq \|Ly - b\|_2\}$ is a second-order cone that can be represented as $\|(Ly - b, s - 0)\|_2 \leq s$ with $s \geq 0$.

Speed-cap hinge (used in §5.1). Let $\mathbf{V} \in \mathbb{R}^{2T\times 2(T+1)}$ pick the velocity rows per step from $\mathbf{D}_v$ so that $v_t = \big((\mathbf{V}_x y)_t, (\mathbf{V}_y y)_t\big) \in \mathbb{R}^2$. Introduce epigraph radii $r_t \geq \|v_t\|_2$ via SOCs and hinge slacks $s_{t,\ell} \geq 0$:

$$\|v_t\|_2 \;\leq\; r_t, \qquad r_t \;\leq\; v_{\max}(t, \xi) - \tau_\ell + s_{t,\ell}, \qquad \forall t,\ \ell.$$

The atom $a_\ell^{\mathrm{spd}}(\xi, y) = \Delta t \sum_t s_{t,\ell}^2$ is then quadratic in the slacks and convex in y.

Lane corridor bands. Given a signed distance linearization $n_t^\top q_t \leq b_t$ (left / right bands), stack them over t to form $\mathbf{G}(\xi)y \leq \mathbf{h}(\xi)$ with block rows $[n_t^\top, 0, \dots, 0, n_t^\top, 0, \dots]$ aligning to q_t.

Distance margins to static polygons. Let $L_t y$ be the local linearization of the closest point map at time t (frozen per inner solve). A margin $\|L_t y - b_t\|_2 \leq u_t$ is SOC: add to the family $\|\mathbf{A}_m(\xi)y - \mathbf{b}_m(\xi)\|_2 \leq u_m(\xi)$ in (5).

F Averaging operators for temporal windows

For a window $W \subset \{0, \dots, T-1\}$, let $S_W \in \mathbb{R}^{2|W|\times 2(T+1)}$ average the entries per axis of a time series aligned with differences ($\mathbf{D}_v$ or $\mathbf{D}_a$). Concretely, for a 1D series $z \in \mathbb{R}^{T'}$ (with $T' = T$ for $\mathbf{D}_v$ or $T' = T{-}1$ for $\mathbf{D}_a$), define $S_W^{(1D)} \in \mathbb{R}^{|W|\times T'}$ by

$$(S_W^{(1D)} z)_\tau = \frac{1}{|W|} \sum_{t\in W} z_t,$$

and set $S_W := \mathrm{blkdiag}(S_W^{(1D)}, S_W^{(1D)})$. All S_W are sparse with at most $2|W|$ nonzeros per row block; they share the grid in4.1.

G Euclidean projection onto the probability simplex

We project $z \in \mathbb{R}^R$ onto the simplex $\Delta_{R-1} = \{w \in \mathbb{R}^R \mid \mathbf{1}^\top w = 1,\ w \geq 0\}$ in $O(R \log R)$ time through classification. The solution to $\min_{w \in \Delta_{R-1}} \frac{1}{2}\|w - z\|_2^2$ has a closed form.

$$w^\star = \max\{\, z - \theta^\star \mathbf{1},\ 0 \,\}, \qquad \theta^\star = \frac{1}{\rho}\Big(\sum_{j=1}^{\rho} \mu_j - 1\Big),$$

where μ is z sorted in descending order, and $\rho = \max\big\{j \in \{1, \ldots, R\} \;\big|\; \mu_j - \frac{1}{j}\big(\sum_{r=1}^{j} \mu_r - 1\big) > 0\big\}$.

Algorithm 4 ProjectToSimplex(z) — Euclidean projection onto Δ_{R-1}

Input: $z \in \mathbb{R}^R$
Sort $\mu \leftarrow \textsc{SortDesc}(z)$; compute partial sums $S_j = \sum_{r=1}^{j} \mu_r$
Find $\rho \leftarrow \max\{\, j \mid \mu_j - \frac{1}{j}(S_j - 1) > 0 \,\}$
$\theta \leftarrow \frac{1}{\rho}(S_\rho - 1)$
return $w = \max\{\, z - \theta,\ 0 \,\}$ (componentwise)

Row-wise matrix projection. When a matrix $W \in \mathbb{R}^{N \times R}$ needs a row-wise projection onto Δ_{R-1} (e.g., pressed w_i), apply Alg. 4 independently to each row. Numerical tip: reuse the sort indices to avoid allocations; clip tiny negatives to zero and renormalize by $\max(\sum w, 1)$ for safety on the 10^{-12} scale.

Usage in our master. At every master step, project both $w_i = \Theta_w \phi(\xi_i)$ and any free w_i variables onto Δ_{R-1} using Alg. 4; this guarantees feasibility of (11) and stabilizes Benders cuts (15).

H Warm-start from Trajecton++

Goal. Given a Trajecton++ candidate $\bar{y}$ (positions) that may not exactly match our grid/constraints, we produce a feasible, solver-friendly warm start $y^{(0)} \in \mathcal{Y}(\xi)$ and (optionally) an initial active-set guess for the inner solver.

H.1 Grid alignment and denoising

If $\bar{y}$ is on a different grid, re-sample to our grid $(\Delta t, T)$ by linear (or Hermite) interpolation on positions, then reconstruct velocity/acceleration/jerk by $(\mathbf{D}_v, \mathbf{D}_a, \mathbf{D}_j)$ (§4.1). Optionally apply a light Tikhonov smoothing:

$$y^{\mathrm{sm}} = \arg\min_{y}\ \tfrac{1}{2}\|y - \bar{y}\|_{W_0}^2 + \tfrac{\kappa}{2}\|\mathbf{D}_a y\|_2^2,$$

with a small κ (for example, 10^{-3}) and diagonal weight W_0 (default identity).

H.2 Equality projection (boundary/initialization)

Project y^{sm} onto the equality set $\mathbf{E}(\xi)y = \mathbf{f}(\xi)$:

$$y^{\mathrm{eq}} = \arg\min_{y} \ \tfrac{1}{2}\|y - y^{\mathrm{sm}}\|_{W_1}^2 \quad \text{s.t.} \quad \mathbf{E}(\xi)y = \mathbf{f}(\xi).$$

The KKT system gives the closed form

$$\begin{bmatrix} W_1 & \mathbf{E}(\xi)^\top \\ \mathbf{E}(\xi) & 0 \end{bmatrix} \begin{bmatrix} y^{\mathrm{eq}} \\ \nu \end{bmatrix} = \begin{bmatrix} W_1 y^{\mathrm{sm}} \\ \mathbf{f}(\xi) \end{bmatrix},$$

where W_1 is diagonal (e.g. I). Factorize once per scene.

H.3 SOC/linear feasibleization (single-shot projection)

Compute a strictly feasible warm start using a single convex projection:

$$\begin{aligned} y^{(0)} = \arg\min_{y} \ & \tfrac{1}{2}\|y - y^{\mathrm{eq}}\|_{W_2}^2 \\ \text{s.t.} \ & \mathbf{G}(\xi)y \le \mathbf{h}(\xi) - \delta\,\mathbf{1}, \\ & \|\mathbf{A}_m(\xi)y - \mathbf{b}_m(\xi)\|_2 \le u_m(\xi) - \delta, \quad \forall m, \end{aligned} \tag{16}$$

with a small interior margin $\delta > 0$ (for example, 10^{-3} in scene units). This is a QP/SOCP (the same cones as the inner problem).

Speed-cap hinge variables (optional pre-activation). To reduce the first iteration work, precompute $r_t = \|(\mathbf{D}_v y^{(0)})[t]\|_2$ and set initial hinge slacks $s_{t,\ell}^{(0)} = \max\{0,\, r_t - (v_{\max}(t,\xi) - \tau_\ell)\}$; pass them to the inner solver as starting values.

H.4 Active-set seed and pools

Mark linear constraints $\mathbf{G}(\xi)y \le \mathbf{h}(\xi)$ active in tolerance ϵ_{act}, and for SOC constraints, mark those with residuals within ϵ_{act} as "nearly active". Store these sets alongside $(Q, g, \hat{w})$ in the active set pool, enabling warm-starts across scenes with similar geometry.

Summary. Steps §H.1–§H.3 yield a feasible $y^{(0)} \in \mathcal{Y}(\xi)$ close to Trajecton++ while strictly interior by δ, improving inner convergence and Benders cut quality without touching the master.

I Proofs for §4

Proof of the Lemma (Danskin + supporting cuts). Let $f(w, y) = \langle w, \tilde{A}(\xi, y)\rangle$, which is linear (hence convex) in w and continuous on $\mathcal{Y}(\xi)$. By A1-A2, $\mathcal{Y}(\xi)$ is non-empty, convex, and compact, and the minimizer set

$\arg\min_{y \in \mathcal{Y}(\xi)} f(\hat{w}, y)$ is non-empty. Danskin's theorem for the pointwise minimum of convex functions gives

$$\partial_w Q(\xi, \hat{w}) \;=\; \operatorname{conv}\big\{\, \nabla_w f(\hat{w}, y) \;\big|\; y \in \arg\min_{u \in \mathcal{Y}(\xi)} f(\hat{w}, u) \,\big\}.$$

Since $\nabla_w f(\hat{w}, y) = \tilde{A}(\xi, y)$, any choice $g = \tilde{A}(\xi, y^\star)$ with $y^\star$ a minimizer belongs to $\partial_w Q(\xi, \hat{w})$ (or more generally any convex combination of such vectors does). The subgradient inequality for a convex function then yields

$$Q(\xi, w) \;\geq\; Q(\xi, \hat{w}) \;+\; g^\top (w - \hat{w}), \quad \forall w,$$

which is exactly the bender cut (outer approximation) in w. If the inner objective includes the Tikhonov term $\frac{\varepsilon}{2}\|y\|_2^2$ (A3), the minimizer is unique and $\partial_w Q(\xi, \hat{w}) = \{\tilde{A}(\xi, y^\star)\}$, but uniqueness is not required for the validity of the cut. □

J Reproducibility details

Environment. Ubuntu 22.04; CPU: *e.g.* Intel Xeon 6226R; RAM: 128 GB; optional GPU unused by solvers.

Solvers and versions. *e.g.* OSQP 0.6.3 / ECOS 2.0.11 for QP/SOCP; CVXPy 1.5 or custom C++ wrappers. All conic problems use identical tolerances as in Table 1.

Randomness. Random seeds: 2025 for data shuffles; 17 for candidate subsampling; 42 for initialization. We fix all NumPy/PyTorch/C++ RNGs accordingly.

Batching and threads. Batch size 8 scenes for offline policy; 1 scene per inner solve; BLAS threads pinned to 4.

Code pointers. We release scripts to reproduce Table 3 and 2, including logs of cut counts and active-set reuse.