GEOMETRIC PARETO CONTROL: PHYSICS-SUPERVISED PARETO REPRESENTATION LEARNING VIA RIEMANNIAN ENERGY-GRADIENT FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

We study multi-objective sequential control problems in physical systems whose dynamics and operational constraints are known or can be represented by accurate physics-based models. Although reinforcement learning has been widely explored in such settings, standard policy learning still faces high-dimensional action search, fixed or externally supplied objective trade-offs, retraining under changed priorities, and hard feasibility requirements at deployment. Our premise is that known physics and constraints expose a compressible action structure that can be learned as a reusable response map before deployment. The resulting Pareto-relevant response family is organized before deployment, so online control begins with a physics-supervised prediction and local correction instead of repeatedly solving a nonlinear program. We propose Geometric Pareto Control (GPC), which embeds the supported family of dynamically feasible Pareto-optimal control responses into a continuous latent homotopy space using offline scalarized optimization under the known dynamics. At deployment, the measured physical state supplies priorities to the learned map; guarded local updates and domain-specific refinement adapt its decoded action or plan. We state formal assumptions under which decoded actions remain feasible and the action error does not accumulate over the rollout horizon. Across analytical control, safe multi-agent navigation, and optimal power flow, GPC compares favorably with optimization and safe-RL baselines in safety, feasibility, and real-time multi-objective performance. These results support physics-supervised learning of reusable Pareto-relevant responses that can be adapted online under changing priorities.

1 INTRODUCTION

Control in power networks and robotics must balance objectives that change with the physical state. A robot approaching a moving obstacle should prioritize clearance; a power network near a voltage or thermal limit should prioritize recovery. Once the danger recedes, progress or economic operation should regain weight. The action must also satisfy coupled physical constraints and arrive within the decision deadline. These requirements arise broadly in learning-based power, transportation, and communication control (Su et al., 2025; Haydari & Yılmaz, 2022; Zong et al., 2025; Luong et al., 2019; Feriani & Hossain, 2021). The challenge is therefore both computational and representational: which useful feasible actions should be available, and how should a controller move among them?

Many such systems already have useful dynamics and constraint models. Repeated online optimization exploits these models directly but pays the solve cost at each decision (Garcia et al., 1989; Bertsekas, 2016; 2017). Amortized optimization instead learns a reusable solution map across related problem instances (Amos, 2023); reinforcement learning similarly amortizes decisions into a policy (Kaelbling et al., 1996; Sutton et al., 1998; Arulkumaran et al., 2017; Bertsekas, 2019), with safety typically handled through constrained training or an additional runtime mechanism (Achiam et al., 2017; Chow et al., 2018; Wabersich et al., 2023). We study a complementary use of the same physics: construct a family of constrained multi-objective solutions offline, represent its local structure, and navigate that representation online. In cyber-physical systems, governing dynamics and tightly coupled operational constraints sharply restrict admissible behavior, naturally compressing Pareto-relevant responses into a low-dimensional geometry.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Geometric Pareto Control (GPC) separates three objects that are often conflated: an offline scalarization weight used to generate diverse responses, a runtime urgency coordinate derived from the current state, and a latent location that represents a candidate response. The offline weight covers alternative trade-offs. The urgency coordinate is a state-dependent feedback signal that converts current operating conditions into objective priorities. The learned response map supplies an initial action, which local feedback and feasibility correction adapt to the latest measurement. The controller thus reuses optimization work without retraining the neural map whenever priorities change.

The contributions are: (i) a physics-supervised solution-map construction with observation and reference encoders, geometric neighborhood regularization, and rollout-aware response-map refinement; (ii) an adaptive feedback layer that recomputes objective priorities from the measured physical state, with guarded RK2/Lie latent-flow correction for tightly constrained optimal power flow (OPF) and learned plan correction with finite refinement for navigation; (iii) explicit deployment guarantees that account for the final output map, coupled constraints, equality residuals, and repeated closed-loop re-localization; and (iv) an uncertainty-transferable, physics-supervised response map that handles unseen parameter realizations within the modeled physical family without retraining. More broadly, GPC provides a framework for determining whether known physics and constraints expose a compressible structure in a cyber-physical system's feasible action space: namely, whether its Pareto-relevant actions admit a low-dimensional representation that supports local online correction or finite plan refinement without full online re-optimization. The empirical study spans three domains: analytical control with two objective landscapes, static and moving multi-agent navigation, and AC power flow under nominal, parameter-perturbed, and topology-shifted conditions.

2 Related Work

Safe and constrained RL uses constrained policy updates, Lyapunov conditions, shields, or model-based safety filters to reduce deployment violations (Achiam et al., 2017; Chow et al., 2018; Alshiekh et al., 2018; Wabersich et al., 2023). GPC instead uses constrained solutions to organize a feasible response family before deployment; its final projection and refinement remain explicit parts of the evaluated controller. Multi-objective and preference-conditioned learning represents reward trade-offs through Pareto sets or preference vectors (Roijers et al., 2013; Van Moffaert & Nowé, 2014; Abels et al., 2019; Reymond et al., 2022), while offline RL learns reusable decisions from stored experience (Kumar et al., 2020; Kostrikov et al., 2022; Fujimoto & Gu, 2021). Physics-informed learning and model predictive control (MPC) exploit known dynamics and constraints (Raissi et al., 2019; Karniadakis et al., 2021; Garcia et al., 1989). These approaches either learn a policy from data or repeatedly solve a model-based control problem online. GPC instead uses constrained offline solutions to identify and represent the compact geometry of feasible, Pareto-relevant actions induced by known physics and constraints, and adapts the decoded response online using state-dependent priorities and local refinement. Appendix B gives the broader literature context, including geometric methods and applications.

3 Methodology

GPC has two offline stages and one online cycle. First, constrained solves supply a family of reference responses and a dual-encoder representation. Second, rollout-aware refinement supervises the response map at intermediate latent queries and, for plan-valued domains, through differentiable multi-step physical rollouts of decoded horizon plans. Online, the observation re-anchors the learned response, physical urgency supplies priorities, and guarded local corrections adapt the decoded action or plan. For the power-flow latent-flow instantiation, the explicit RK2/Lie update below provides the geometric local step. The evaluated navigation controller uses the same response-map framework with learned plan correction and finite candidate refinement.

3.1 Problem, Inputs, and Offline Reference Generation

Let s_t, p_t, and u_{t-1} denote physical state, operating parameters, and the previous action. The context is $c_t = (s_t, p_t, u_{t-1})$. The available input x_t contains measured state components, task information and known parameters; the previous action is retained when constraints have temporal coupling. GPC outputs either one action or a plan $U = (u_0, \ldots, u_{H-1})$, of which only the first action is exe-

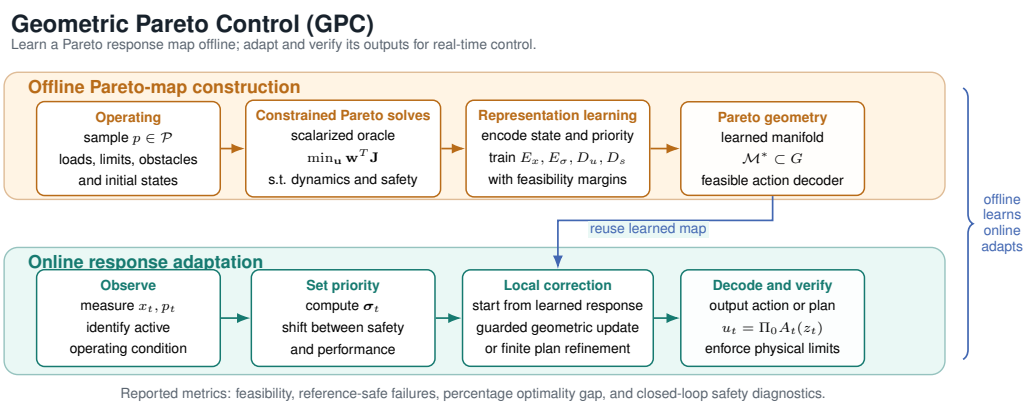


Figure 1: Overview of Geometric Pareto Control (GPC). The offline stage constructs a feasible Pareto geometry $\mathcal{M}^*$ from constrained multi-objective solves, while the online stage uses the learned response map to set priorities, apply local correction, and verify the final control response.

cuted. Let $\mathbf{j}(s_h, u_h, p_h) \in \mathbb{R}^{n_J}$ be the stage-objective vector and define the response-level objective $\mathbf{J}(c, U) = \sum_{h=0}^{H-1} \mathbf{j}(s_h, u_h, p_h)$. For a fixed scalarization weight $w \in \Delta^{n_J-1}$, reference responses solve

$$\begin{aligned} U^\star(w; c) \in \arg\min_{U} \quad & w^\top \mathbf{J}(c, U) \\ \text{s.t.} \quad & s_{h+1} = F_{\text{phys}}(s_h, u_h, p_h), \quad g_c(U) = 0, \quad f_c(U) \le 0. \end{aligned} \tag{1}$$

The constraints include physical, box, and inter-step limits. A single-action domain uses $H = 1$. We sample contexts and scalarization weights, solve (1), and retain successful numerical responses U^{ref}. Positive-weight global solves recover supported Pareto points; numerical nonconvex references inherit their solver's accuracy and local-optimality limits. Each retained record contains

$$\mathcal{D} = \{(x_k, s_k, p_k, u_{k-1}, w_k, \sigma_k, \mathbf{J}_k, U_k^{\text{ref}}, \boldsymbol{\gamma}_k)\}_{k=1}^{N}, \tag{2}$$

where $\boldsymbol{\gamma}_k$ records the separate physical margins and σ_k is computed from physical urgency by (9). The sampling weight w_k covers trade-offs; the urgency vector σ_k is a state-dependent feedback signal encoding the current operating priority. Feasible targets can have zero margin, so retention alone does not establish a positive certified tube.

3.2 Stage I: Dual-Path Representation Learning

An observation encoder E_x provides the online location, while a reference encoder E_o receives richer offline information:

$$z_k^x = E_x(x_k), \qquad z_k^o = E_o(s_k, \sigma_k, \mathbf{J}_k), \qquad \hat{s} = D_s(z), \quad \hat{U} = D_U(z). \tag{3}$$

Both paths share D_s; D_U reconstructs a single action or a horizon plan. With $\langle\cdot\rangle$ denoting the training average, the representation losses $\mathcal{L}_{\text{I}} = \lambda_U \mathcal{L}_U + \lambda_s \mathcal{L}_s + \lambda_c \mathcal{L}_c + \lambda_n \mathcal{L}_n$ are

$$\begin{aligned} \mathcal{L}_U &= \langle \| D_U(z_k^x) - U_k^{\text{ref}} \|^2 \rangle, & \mathcal{L}_c &= \left\langle 1 - \frac{\langle z_k^x, z_k^o \rangle}{\max(\|z_k^x\| \|z_k^o\|, \varepsilon_z)} + \beta \|z_k^x - z_k^o\|^2 \right\rangle, \\ \mathcal{L}_s &= \left\langle \sum_{b \in \{x,o\}} \| D_s(z_k^b) - s_k \|^2 \right\rangle, & \mathcal{L}_n &= \left\langle \sum_{j \in \mathcal{N}_\kappa(k)} e^{-\|p_k - p_j\|^2/\sigma_p^2} \|z_k^x - z_j^x\|^2 \right\rangle. \end{aligned} \tag{4}$$

The reconstruction terms retain action and physical information; alignment transfers the reference path's organization to the deployment encoder; the neighborhood term preserves local continuity across operating conditions. $\varepsilon_z > 0$ stabilizes alignment. E_o is discarded online. When an observation admits several reference responses, E_x supplies an initialization rather than identifying every preference-conditioned solution; the online priority and response adjustment resolve the trade-off.

3.3 Stage II: Rollout-Aware Response Refinement

Stage II exposes the response map to intermediate latent queries and, for plan-valued outputs, evaluates the decoded plan through multiple physical steps. We keep E_o and D_s frozen. The transition-indexed instantiation also freezes E_x and refines only D_U; a plan-valued instantiation may jointly refine E_x, D_U, and auxiliary response heads. For a stored transition, let $\Delta U_h^{\mathrm{ref}} = U_{h+1}^{\mathrm{ref}} - U_h^{\mathrm{ref}}$ and $z_h^\alpha = (1-\alpha)z_h + \alpha z_{h+1}$, with response-target and flow fractions $\alpha_r, \alpha_f \in (0,1]$; the flow target is $U_h^{\mathrm{flow}} = U_h^{\mathrm{ref}} + \alpha_f \Delta U_h^{\mathrm{ref}}$. Define the direct step $d_E = z_h^{\alpha_r} - z_h$ and the Lie-corrected query $\tilde{z}_{h+1} = z_h \oplus (d_E + \gamma_L d_L)$, where d_L is obtained from the local decoder metric and Lie generator detailed in Appendix F. Unlike z_{h+1}, which encodes the next reference, $\tilde{z}_{h+1}$ is the fast, solver-free local feasibility correction. The displacement guard accepts it only inside the certified neighborhood; under Theorem C.1, the decoded response then remains in the feasibility tube.

For a rollout depth $1 \le K \le H$, each query-target pair carries $\mathcal{R} = (s^{\mathrm{tar}}, p_{0:K}^{\mathrm{tar}}, s_{1:K}^{\mathrm{ref}}, G_{1:K}^{\mathrm{ref}})$: an aligned initial state, parameter sequence, and reference rollout. Given a query z, decode a horizon plan and propagate its first K actions through the known physics:

$$\begin{aligned} U(z) &= D_U(z) = (u_0(z), \dots, u_{H-1}(z)), \qquad \hat{s}_0 = s^{\mathrm{tar}}, \\ \hat{s}_{k+1} &= F_{\mathrm{phys}}(\hat{s}_k, u_k(z), p_k^{\mathrm{tar}}), \qquad \hat{G}_{k+1} = \Gamma(\hat{s}_{k+1}, p_{k+1}^{\mathrm{tar}}), \quad 0 \le k < K, \end{aligned} \tag{5}$$

Here Γ constructs the interaction representation. Let d_s and d_G be nonnegative discrepancy functions, let $m_j(s,G) \ge 0$ indicate constraint satisfaction, and set $[a]_+ = \max(a,0)$. The loss for one query uses the same decoded plan in its response and rollout terms:

$$\begin{aligned} \Phi_K(z, U^{\mathrm{tar}}; \mathcal{R}) = \lambda_U \|U(z) - U^{\mathrm{tar}}\|^2 + \lambda_R \sum_{k=1}^{K} \Big[\lambda_s d_s(\hat{s}_k, s_k^{\mathrm{ref}}) + \lambda_G d_G(\hat{G}_k, G_k^{\mathrm{ref}}) \\ + \lambda_{\mathrm{mar}} \sum_j [-m_j(\hat{s}_k, \hat{G}_k)]_+^2 \Big]. \end{aligned} \tag{6}$$

Thus Φ_K directly couples a latent query to the physical trajectory generated by its decoded response. Let $\mathcal{R}_h$ and $\mathcal{R}_{h+1}$ be the endpoint-aligned records, let $\mathcal{R}_h^{\mathrm{flow}}$ be an offline-aligned flow record, and let ω_h be the normalized transition weight defined in Appendix F. Applying Φ_K to the endpoint, next-target, flow, and Lie-corrected supervision pairs gives

$$\begin{aligned} \mathcal{L}_{\mathrm{II}}^{(K)} = \Big\langle \omega_h \Big[\Phi_K(z_h, U_h^{\mathrm{ref}}; \mathcal{R}_h) + \Phi_K(z_{h+1}, U_{h+1}^{\mathrm{ref}}; \mathcal{R}_{h+1}) + \lambda_r \Phi_K(z_h^{\alpha_r}, U_{h+1}^{\mathrm{ref}}; \mathcal{R}_{h+1}) \\ + \lambda_f \Phi_K(z_h^{\alpha_f}, U_h^{\mathrm{flow}}; \mathcal{R}_h^{\mathrm{flow}}) + \lambda_L \Phi_K(\tilde{z}_{h+1}, U_{h+1}^{\mathrm{ref}}; \mathcal{R}_{h+1}) \Big] \Big\rangle. \end{aligned} \tag{7}$$

If $\mathcal{R}_h^{\mathrm{flow}}$ is unavailable, the flow branch keeps only $\lambda_U \|U(z_h^{\alpha_f}) - U_h^{\mathrm{flow}}\|^2$. Optional increment and plan-smoothness regularizers are given in Appendix F. All weights are nonnegative, and $\omega_h = 1$ when transition weighting is unused. The choice $K = 1$ recovers one-step training; navigation uses $K = H$ and refines E_x, D_U, and its safety head.

Algorithm 1 Offline construction and two-stage training

Require: Known dynamics/constraints, context distribution, scalarization weights, model modules

1: Sample contexts and weights; solve (1); retain complete successful response chains
2: Record observations, states, reference plans, objectives, urgency labels, and physical margins
3: Train E_x, E_o, D_s, D_U jointly with stage-I loss (4)
4: Form aligned rollouts and per-query losses by (5)–(6)
5: Optimize the active modules with the unified objective (7)
6: Validate response error and constraints on held-out decoded rollouts
7: Freeze the deployed modules; retain calibration data for localization and margin checks

3.4 Observation Reset, Urgency, and the Energy Field

Every cycle starts from $z_{\mathrm{obs,t}} = E_x(x_t)$. For the measured-state component x_t^{obs} and observation map h_{obs}, define the closed context-specific reference fiber $\mathcal{M}_t \subseteq \mathcal{M}^*$ by

$$\begin{aligned} \mathcal{M}_t &= \{ z \in \mathcal{M}^* : \|x_t^{\mathrm{obs}} - h_{\mathrm{obs}}(D_s(z))\|^2 \le \tau_t \}, \quad \tau_t = \hat{\nu}_t + \tau_{\mathrm{geom}}, \\ z_t^0 &= (1-\alpha_{\mathrm{loc}}) z_{t-1}^+ + \alpha_{\mathrm{loc}} \, \mathrm{proj}_{\mathcal{M}_t}(z_{\mathrm{obs,t}}). \end{aligned} \tag{8}$$

$\hat{\nu}_t$ estimates squared measurement-noise energy and τ_{geom} is a squared-residual allowance calibrated from held-out reconstruction errors, so both terms have the units of the squared norm. Here $\alpha_{\text{loc}} \in (0, 1]$, and $\text{proj}_{\mathcal{M}_t}$ denotes nearest-point localization on the current reference fiber; direct encoder reset uses $z_t^0 = z_{\text{obs,t}}$. Smoothing reduces noise without certifying membership. The tube theorem requires the localization error, including any approximate projection and smoothing. Reconstruct $\hat{s}_t = D_s(z_t^0)$ and compute normalized indicators $\delta_i(\hat{s}_t, p_t) \in [0, 1]$. For gains $k_i \geq 0$, scales $\epsilon_i > 0$, offset $\rho > 0$, filter coefficient $\alpha_\sigma \in (0, 1]$, and initialization $\sigma_0 \in \Delta^{n_J-1}$, the priority update is

$$\phi_{t,i} = \exp(k_i \delta_i(\hat{s}_t, p_t)/\epsilon_i) - 1, \qquad \bar{\sigma}_{t,i} = \frac{\phi_{t,i} + \rho}{\sum_j (\phi_{t,j} + \rho)}, \qquad \sigma_t = (1 - \alpha_\sigma)\sigma_{t-1} + \alpha_\sigma \bar{\sigma}_t. \tag{9}$$

The optional low-pass filter suppresses chattering; $\alpha_\sigma = 1$ gives the unfiltered rule. In OPF, indicators measure maximum normalized loading and voltage deviation, with a small positive economic background. Their scales set the nominal priority; equal zero indicators give balanced weights. Holding the current measurement and σ_t fixed during a step, define, for restoration scale $\epsilon_{\text{snap}} > 0$,

$$V_t(z) = \epsilon_{\text{snap}}^{-1} \|x_t^{\text{obs}} - h_{\text{obs}}(D_s(z))\|^2 + \sum_i \sigma_{t,i} J_i(D_s(z), D_U(z), p_t),$$

$$\mathcal{F}_t(z) = -M(z)^{-1} \nabla_z^\circ V_t(z), \qquad M(z) \succ 0. \tag{10}$$

The first term restores measurement consistency ("snap"); the second drives priority-weighted improvement ("slide"). They describe the intended behavior, without assuming exact normal/tangent decomposition. Here ∇° denotes the ordinary gradient at differentiable points and the implementation's selected autodifferentiation subgradient at kinks introduced by max, absolute-value, or clipping operations; smooth-field RK2 accuracy statements apply away from those kink sets. The choice $M = I$ gives the Euclidean field; the metric-preconditioned implementation uses $M = G$ as defined below. Differentiation passes through frozen decoders; online neural weights are unchanged. At a smooth point, this derivative makes the physical feedback path explicit. Let $r_t(z) = h_{\text{obs}}(D_s(z)) - x_t^{\text{obs}}$ and evaluate all Jacobians at z. Then

$$\nabla_z V_t(z) = 2\epsilon_{\text{snap}}^{-1} J_{D_s}^\top J_{h_{\text{obs}}}^\top r_t(z) + \sum_i \sigma_{t,i} \left[J_{D_s}^\top \nabla_s J_i + J_{D_U}^\top \nabla_U J_i \right]. \tag{11}$$

Thus measurement residuals and physical objective derivatives are pulled back through the frozen decoders into one latent direction. The preconditioner $M(z)^{-1}$ changes the local geometry of that direction but not the feedback variables held fixed during the within-cycle step.

The following explicit latent-flow update locally refines a response supplied by the learned map; it does not replace the offline construction of that map.

3.5 Capped RK2, Lie-Local Correction, and Execution

Let $\overline{\mathcal{F}}_t = \text{proj}_{\|v\| \leq V_{\max}} \mathcal{F}_t$, and distinguish latent step h_z from physical sampling interval Δt. The explicit midpoint predictor, a second-order Runge–Kutta (RK2) step, is

$$k_1 = \overline{\mathcal{F}}_t(z_t^0), \quad z_{1/2} = z_t^0 + \tfrac{1}{2} h_z k_1, \quad k_2 = \overline{\mathcal{F}}_t(z_{1/2}), \quad d_E = h_z k_2. \tag{12}$$

For $\lambda_m > 0$, the decoder-pullback metric $G = J_{D_s}^\top J_{D_s} + \lambda_m I$ is positive definite. Let B_G contain its top-k orthonormal eigenvectors and let $\boldsymbol{\lambda}$ contain the associated positive eigenvalues, following standard local-metric constructions in Riemannian optimization (Absil et al., 2008; Boumal, 2023; De Bortoli et al., 2022). With $d = B_G^\top d_E$ and $v = (B_G^\top k_2) \oslash \sqrt{\boldsymbol{\lambda}}$, rotate each three-dimensional block by $R_b = \exp(s_L h_z [v_b]_\times)$ and form

$$d_L = B_G \left[\text{blockdiag}(R_b) d - d \right], \quad \bar{z} = z_t^0 \oplus (d_E + \gamma_L d_L), \quad z_t^+ = (1 - \mu_R)\bar{z} + \mu_R \widehat{\text{proj}}_{\mathcal{M}^*}(\bar{z}). \tag{13}$$

$[v_b]_\times$ is skew-symmetric; $\oplus$ denotes the chosen latent group composition, or addition in a Euclidean chart (Lee, 2018; Murray et al., 2017; Barfoot, 2024). A final two-dimensional block is zero-padded to three dimensions, rotated, then truncated; one-dimensional blocks are unchanged. A non-finite residual is replaced by zero. $\gamma_L = \mu_R = 0$ recovers the plain RK2 update. The metric enters the residual and, when $M = G$, the field preconditioner. Group closure does not establish physical feasibility, and the accepted displacement includes the reference correction.

Remark (online computation). Equations (12)–(13) require two evaluations of $\overline{\mathcal{F}}_t$, local latent-space linear algebra through $M(z)$ and B_G, and small-block rotations. The reference correction is the indexed query $\widehat{\text{proj}}_{\mathcal{M}^*}(\bar{z}) = z_{j^\star}^o$, where $j^\star = \arg\min_{1 \leq j \leq N} \|\bar{z} - z_j^o\|$; thus each cycle uses a fixed-depth update and one fast lookup rather than re-solving (1).

Algorithm 2 GPC deployment with the explicit geometric navigator

Require: Observation x_t, parameters p_t, previous action u_{t-1}, frozen trained map
1: Encode the latest observation and obtain z_t^0 by (8) or direct reset
2: Reconstruct $\hat{s}_t$; compute current physical indicators and priority (9)
3: Compute the two capped slopes (12), decoder metric, and guarded update (13)
4: Measure $\|z_t^+ - z_t^0\|$; backtrack or reject if a configured displacement guard is exceeded
5: Decode $U_t \leftarrow D_U(z_t^+)$; enforce limits and check feasibility
6: Execute $u_t = \Pi_0 U_t$; retain the actual action and acquire the next observation

4 A Conditional Deployment Tube

A deployment tube transfers latent localization and update errors to physical feasibility and action error through decoder sensitivity and certified margins. For context c_t, let $A_t(z)$ denote the final response used to evaluate constraints, $\mathcal{M}_t \subseteq \mathcal{M}^*$ the nonempty closed reference fiber in (8), and $y_t^{\mathrm{ref}}(a)$ a matched response for $a \in \mathcal{M}_t$, not necessarily a global optimum. Let Π_t be the nonexpansive projection to the executed action, $f_{t,i}$ the non-box inequalities, and g_t the equality residuals. Appendix Theorem C.1 proves the one-cycle certificate. Re-localizing at every cycle yields Theorem 1: uniform margins preserve feasibility over any finite rollout and give a horizon-independent matched-action error bound. Proposition 4 derives these constants from a verified finite cover.

Theorem 1 (Closed-loop feasibility and non-cumulative action error)**.** *Suppose the conditions of Appendix Theorem C.1 hold at every closed-loop cycle with common constants, and the controller re-localizes from the current measurement before every accepted update. Let δ_{step} bound the total accepted latent displacement, define $r := \delta_{\mathrm{loc}} + \delta_{\mathrm{step}}$, and choose $a_t \in \arg\min_{a\in\mathcal{M}_t} \|z_t^+ - a\|$. If*

$$L_i K r \le \gamma_i \quad \forall i, \qquad \eta_0 + L_g K r \le \eta, \tag{14}$$

then, for every rollout horizon $T \ge 1$, the final response is feasible at every cycle:

$$f_{t,i}(A_t(z_t^+)) \le 0 \quad \forall i, \qquad \|g_t(A_t(z_t^+))\| \le \eta, \qquad 0 \le t < T. \tag{15}$$

Moreover, let $u_t = \Pi_t A_t(z_t^+)$ and $u_t^{\mathrm{ref}} = \Pi_t y_t^{\mathrm{ref}}(a_t)$. Then

$$\sup_{0\le t<T} \|u_t - u_t^{\mathrm{ref}}\| \le K(\delta_{\mathrm{loc}} + \delta_{\mathrm{step}}) + \delta_{\mathrm{dec}}. \tag{16}$$

Thus the executed-action error is non-cumulative: its bound is independent of T rather than growing across the rollout.

Proposition 1 (Accepted-step accounting for the geometric update)**.** *Suppose the cap gives $\|d_E\| \le h_z V_{\max}$, the block-exponential residual satisfies $\|d_L\| \le C_\omega V_{\max}^2 h_z^2$, and the implemented composition/retraction obeys $\|\mathrm{Retr}_z(v) - z\| \le C_R\|v\|$ on the accepted chart. If a later latent refinement moves the candidate by at most δ_{refine}, then a valid step bound is*

$$\delta_{\mathrm{step}} \le C_R\big(h_z V_{\max} + \gamma_L C_\omega V_{\max}^2 h_z^2\big) + \delta_{\mathrm{refine}}. \tag{17}$$

The first term is the capped predictor, the second is the locally quadratic geometric correction, and the last accounts for any subsequent accepted latent move. Appendix Proposition 11 and the immediately following refinement argument give the proof and regularity conditions; without those assumptions, the directly measured total accepted displacement remains the appropriate quantity.

5 Experiments

We evaluate response quality, physical constraint satisfaction, and decision cost across three domains: analytical control, multi-agent navigation, and AC power flow. The main stress tests are multi-agent navigation and perturbed AC power flow. Analytical convex/nonconvex objectives and nominal OPF provide complementary reference comparisons, while the appendix reports a held-out topology-shift study. The navigation study uses 100 evaluation episodes per setting; OPF and analytical aggregates are taken from their benchmark summaries.

Table 1: Navigation results. The full-information columns compare aggregate GPC and planner costs on matched feasible scenes; cost difference is $100(\sum J_{\mathrm{GPC}}/\sum J_{\mathrm{Planner}} - 1)$. †: results are conditioned on reachability; the one NMPC-unreachable scene is reported in the Appendix.

		Reliability (%)			Full-information comparison	
Scene	Method	Joint	Collision-free	Safe success	Total cost	Cost diff. (%)
Static	**GPC**	100	100	100		
	Multistart NMPC	95	62	60	**GPC**: 24,349.54	+2.81
	PPO / PPO-Lagrangian	0 / 0	5 / 5	0 / 0	Planner: 23,683.80	
	CPO / PPO-shielded	0 / 0	13 / 8	0 / 0		
Moving	**GPC**	99†	100	99†		
	Multistart NMPC	83	42	35	**GPC**: 18,201.83	+1.85
	PPO / PPO-Lagrangian	0 / 0	0 / 0	0 / 0	Planner: 17,871.48	
	CPO / PPO-shielded	0 / 0	0 / 0	0 / 0		

5.1 Multi-Agent Navigation: Coupled Scene Safety

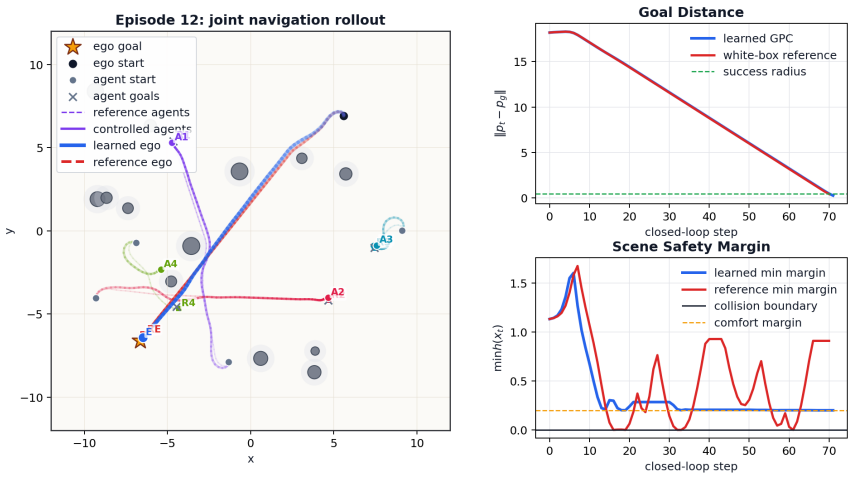
The benchmark controls an ego robot and up to five neighbors among as many as 15 static or moving obstacles. It uses midpoint unicycle dynamics over an eight-step horizon with $\Delta t = 0.2$ s, enforcing control, slew, robot–obstacle, and robot–robot constraints; six objectives represent progress, clearance, effort, violations, terminal error, and smoothness. The learned response map supplies a joint plan, which is corrected, projected to the box and slew limits, and refined by a finite structured candidate search. Joint success requires every active robot to reach its goal, collision freedom requires nonnegative scene-safety margins, and safe success requires both. We compare against multistart nonlinear MPC (NMPC) with known dynamics and constraints, PPO (Schulman et al., 2017), PPO-Lagrangian (Tessler et al., 2019), CPO (Achiam et al., 2017), and shielded PPO (Alshiekh et al., 2018). Appendix I gives the equations, dataset sizes, metric definitions, and reference conditioning; Appendix F.5 specifies candidate construction, search settings, and feasibility-first ranking.

Results. GPC obtains 100% collision-free rollouts in both settings, with 100% static safe success. The moving safe-success summary is 99% under reachability conditioning. Multistart NMPC succeeds safely in 60% of static and 35% of moving episodes. PPO, PPO-Lagrangian, CPO, and one-step shielded PPO all have 0% safe success under this protocol. The online navigation decision takes approximately 13.5 ms, within the 200 ms control interval. These results demonstrate the challenge of simultaneously maintaining whole-scene clearance and reaching every goal; they do not establish a universal limitation of safe RL. Figure 2 makes both progress and margin evolution visible. NMPC receives the known dynamics, constraints, and moving-obstacle forecast over its horizon, but its finite multistart budget does not remove the nonconvexity of the online problem.

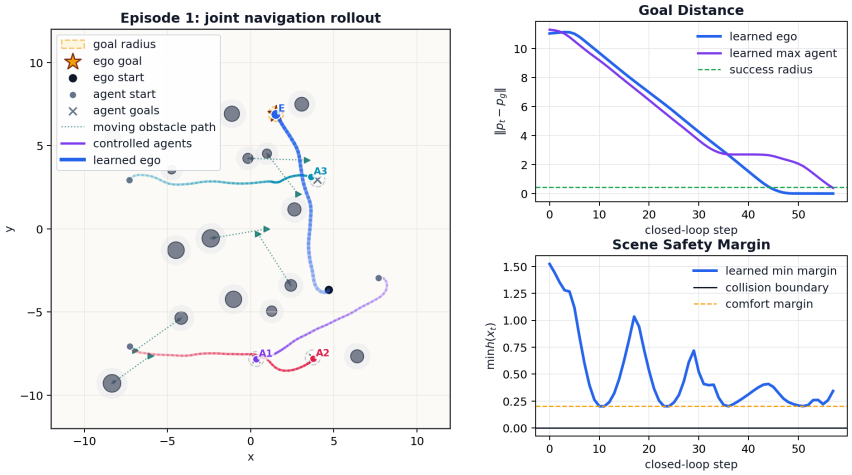
To separate finite-horizon NMPC failures from solution quality, we also compare GPC with a full-information white-box planner on matched scenes in Table 1. The planner jointly refines the full control trajectory, initialized from the corresponding GPC rollout; in the moving scene it is given the exact future obstacle trajectory, including boundary reflections. Both methods safely reach the goal throughout the paired comparison.

Table 2: OPF benchmark results for 300-step rollouts. Latency is mean decision time; perturbed-run objective means use 299 finite costs. Feasibility includes generator limits. Lower cost and latency, and higher feasibility, are desirable.

Network	Method	Mean total objective	Feasible (%)	Latency (ms)
Nominal	**GPC**	361.08	100.00	12.3
	MPC	360.99	100.00	1193.8
	CPO	519.78	0.00	0.19
	TD3	743.77	0.00	0.17
Perturbed	**GPC**	295.30	99.67	12.6
	MPC	295.23	99.67	1228.7
	CPO	383.37	0.00	0.18
	TD3	565.04	0.00	0.16

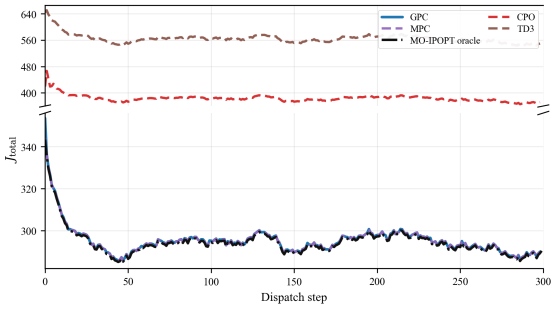


(a) Static scene

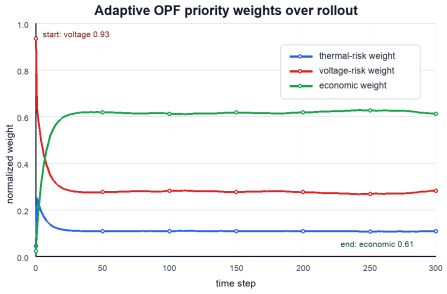


(b) Moving scene

Figure 2: Representative joint trajectories, goal distances, and safety margins. Positive scene margins show the safety behavior of these rollouts. Additional moving scenes appear in Appendix I.4.



(a) Objective trajectory



(b) Adaptive priorities

Figure 3: Perturbed-OPF rollout: total objective and urgency weights. The weights move from recovery toward economic operation. The objective trace shows the operating-cost evolution.

5.2 AC Optimal Power Flow: Performance Under Parameter Variation

Task and information. The IEEE 30-bus task balances thermal recovery, voltage regulation, and generation cost subject to AC balance, equipment limits, and inter-step ramps. We evaluate nominal

and branch-admittance-perturbed 300-step rollouts. The perturbation test evaluates deployment on a held-out realization without retraining. GPC is trained over the perturbation family and receives its descriptor; MPC uses the realized network; TD3 (Fujimoto et al., 2018) and CPO (Achiam et al., 2017) retain nominal training. The comparison is interpreted under these stated training and information protocols. Appendix J gives the sampling, perturbation, reference-filtering, and physical details. The IEEE 118-bus topology study in Appendix J.5 uses 1,001 training graphs spanning $N-0$ through $N-5$ and disjoint 50-graph validation and test sets; its detailed online rollout is on a held-out $N-5$ graph.

Results. Table 2 shows similar objective values for GPC and MPC, with lower GPC latency in both network conditions. Both have one infeasible step in the perturbed run, giving 99.67% feasibility. GPC is $97.1\times$ faster than MPC nominally (12.3 versus 1193.8 ms) and $97.5\times$ faster under perturbation (12.6 versus 1228.7 ms). The perturbed rollout tests transfer to a branch-parameter realization unseen during GPC training but drawn from its modeled perturbation family, without retraining its response map. The nominal-trained RL baselines have no feasible dispatches under the full evaluation constraints in both runs, characterizing their deployment without adaptation after a parameter change; their shorter inference times therefore do not establish a comparable feasible operating point. Figure 3 illustrates adaptive priority evolution, while Appendix J supplies the physical equations and nominal trajectories.

5.3 Analytical Reference Comparisons

The analytical benchmark uses a double integrator with a moving elliptical obstacle and either convex or nonconvex objectives. A discretized finite-horizon dynamic-programming reference evaluates approximation quality over 100 trajectories. GPC's small mean reference gaps coexist with a 2% failure rate in each variant and a 0.05% obstacle-violation fraction in the nonconvex variant. These outcomes are reported separately from cost and computation in Table 3, detailed in Appendix K.

Table 3: Analytical comparison over 100 trajectories per objective. Gap is mean $\pm$ s.e.m. (%); failure is a trajectory-level indicator distinct from reference quality.

Objective	Method	Reference gap (%)	Failure (%)	Time (ms)
Convex	**GPC**	0.043 ± 0.011	2	0.78
	MPC	9.93 ± 0.51	0	1025.24
	Online scalarization	7.90 ± 0.35	67	0.03
	TD3	8.95 ± 0.40	99	0.11
	CPO	382.55 ± 9.91	0	0.10
Nonconvex	**GPC**	0.050 ± 0.011	2	0.97
	MPC	9.43 ± 0.52	0	1229.62
	Online scalarization	7.90 ± 0.38	79	14.65
	TD3	15.11 ± 0.68	100	0.21
	CPO	370.63 ± 9.59	0	0.24

Table 3 separates reference quality from completion. GPC has the smallest gaps in both landscapes. CPO completes every evaluated rollout but remains far from the dynamic-programming reference; TD3 and online scalarization have high failure rates, whereas MPC avoids failures at substantially greater decision cost. Appendix Figure 8(c,g) shows representative tracking, and the nonconvex objective leaves the GPC gap scale essentially unchanged.

6 Conclusion

GPC learns a reusable geometry of feasible, Pareto-relevant responses from constrained solutions and adapts its decoded responses online as priorities change. Under the stated regularity and margin conditions, repeated re-localization preserves closed-loop feasibility and yields a horizon-independent action-error bound. Across analytical control, safe multi-agent navigation, and optimal power flow, GPC demonstrates favorable safety, feasibility, and real-time multi-objective performance relative to optimization and safe-RL baselines. Its principal limitation is structural: as physics and active constraints cease to concentrate Pareto-relevant actions near a low-dimensional geometry, coverage requirements grow and local correction becomes less effective. Overall, the results support GPC's premise that physics-supervised learning can organize reusable Pareto-relevant responses for online correction and refinement without repeated full nonlinear solves.

Reproducibility Statement

The model construction and training procedure are specified in Sections 3.2–3.3 and Appendix F. Complete assumptions and proofs for the deployment guarantees appear in Appendices C–G. Benchmark dynamics, constraints, data generation, evaluation protocols, and metric definitions are documented in Appendices I–K.

AI Use Statement

Generative AI tools assisted manuscript restructuring and editing, formulation and checking of mathematical statements and proofs, and LaTeX and figure layout preparation for this version.

References

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International Conference on Machine Learning*, pp. 11–20. PMLR, 2019.

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31. PMLR, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Eitan Altman. Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical methods of operations research*, 48(3):387–417, 1998.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Brandon Amos. Tutorial on amortized optimization. *Foundations and Trends in Machine Learning*, 16(5):592–732, 2023.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE signal processing magazine*, 34(6):26–38, 2017.

Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2024.

Dimitri Bertsekas. *Nonlinear programming*. Athena Scientific, 2016.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2017.

Dimitri Bertsekas. *Reinforcement learning and optimal control*, volume 1. Athena Scientific, 2019.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*, volume 49. Springer, 2005.

Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. Safe reinforcement learning via shielding under partial observability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14748–14756, 2023.

Kok Wah Chang and Frederick A Howes. *Nonlinear singular perturbation phenomena: theory and applications*. Springer Science & Business Media, 2012.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34:15084–15097, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018a.

Steven Chen, Kelsey Saulnier, Nikolay Atanasov, Daniel D. Lee, Vijay Kumar, George J. Pappas, and Manfred Morari. Approximating explicit model predictive control using constrained neural networks. In *Annual American Control Conference (ACC)*, pp. 1520–1527, 2018b.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in Neural Information Processing Systems*, 31, 2018.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002.

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Amal Feriani and Ekram Hossain. Single and multi-agent deep reinforcement learning for ai-enabled wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials*, 23(2):1226–1252, 2021.

Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. Predicting AC optimal power flows: Combining deep learning and Lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 630–637, 2020.

Vincent François-Lavet, Guillaume Rabusseau, Joelle Pineau, Damien Ernst, and Raphael Fonteneau. On overfitting and asymptotic bias in batch reinforcement learning with partial observability. *Journal of Artificial Intelligence Research*, 65:1–30, 2019.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:20132–20145, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1587–1596, 2018.

Ted Fujimoto, Joshua Suetterlein, Samrat Chatterjee, and Auroop Ganguly. Assessing the impact of distribution shift on reinforcement learning performance. In *NeurIPS 2023 Workshop on Regulatable ML*, 2023.

Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

Javier García and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553): 452–459, 2015.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Ammar Haydari and Yasin Yılmaz. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):11–32, 2022.

Michael Hertneck, Johannes Köhler, Sebastian Trimpe, and Frank Allgöwer. Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2(3):543–548, 2018.

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations*, 2020.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, 34:1273–1286, 2021.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

Benjamin Karg and Sergio Lucia. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, 50(9):3866–3878, 2020.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

Daniel Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pp. 1–6. IEEE, 1987.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

John M Lee. *Introduction to topological manifolds*. Springer, 2000.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pp. 1–29. Springer, 2003.

John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2022.

Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE communications surveys & tutorials*, 21(4):3133–3174, 2019.

Dimitris Michailidis, Willem Röpke, Diederik M Roijers, Sennay Ghebreab, and Fernando P Santos. Scalable multi-objective reinforcement learning with fairness guarantees using lorenz dominance. *Journal of Artificial Intelligence Research*, 85, 2026.

Thomas M Moerland, Joost Broekens, Aske Plaat, and Catholijn M Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, 2017.

Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the Pareto front with hypernetworks. In *International Conference on Learning Representations*, 2021.

Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. DeepOPF: A deep neural network approach for security-constrained DC optimal power flow. *IEEE Transactions on Power Systems*, 36(3):1725–1735, 2021.

Seonho Park and Pascal Van Hentenryck. Self-supervised primal-dual learning for constrained optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4052–4060, 2023.

Matteo Pirotta, Simone Parisi, and Marcello Restelli. Multi-objective reinforcement learning with continuous Pareto frontier approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1110–1118, 2022.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Tong Su, Tong Wu, Junbo Zhao, Anna Scaglione, and Le Xie. A review of safe reinforcement learning methods for modern power systems. *Proceedings of the IEEE*, 2025.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT Press, 1998.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019.

Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained Markov decision processes. In *International Conference on Machine Learning*, pp. 9797–9806. PMLR, 2020.

Yujia Wang and Zhe Wu. Control Lyapunov-barrier function-based safe reinforcement learning for nonlinear optimal control. *AIChE Journal*, 70(3):e18306, 2024.

Lizhi Yang, Blake Werner, Massimiliano de Sa, and Aaron D Ames. CBF-RL: Safety filtering reinforcement learning in training with control barrier functions. *arXiv preprint arXiv:2510.14959*, 2025.

Ahmed S. Zamzam and Kyri Baker. Learning optimal solutions for extremely fast AC optimal power flow. In *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2020.

Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

Zefang Zong, Jingwei Wang, Tao Feng, Tong Xia, Depeng Jin, and Yong Li. Deep reinforcement learning for demand-driven services in logistics and transportation systems: A survey. *ACM Transactions on Knowledge Discovery from Data*, 19(4):1–42, 2025.

A Supplementary Material

The appendix supplies extended related work, the complete deployment proof, certification construction, training and integration details, and benchmark formulations. Additional mathematics supports the main theorem rather than introducing a separate global-optimality claim.

B Extended Related Work

Sequential, safe, and constrained learning. Classical and deep reinforcement learning frame sequential decision-making as policy learning from interaction (Kaelbling et al., 1996; Sutton et al., 1998; Arulkumaran et al., 2017; Bertsekas, 2019). Safe and constrained variants formulate constrained MDPs, trust-region updates, Lyapunov conditions, and safe exploration rules (Altman, 1998; 2021; Achiam et al., 2017; Chow et al., 2018; Wachi & Sui, 2020; García & Fernández, 2012; 2015). Runtime shields, control-barrier constructions, and model-based filters instead modify or certify proposed actions during deployment (Alshiekh et al., 2018; Wang & Wu, 2024; Yang et al., 2025; Wabersich et al., 2023; Carr et al., 2023). GPC is complementary: constrained solves define its response family before deployment, while the deployed map, refinement, and final feasibility checks remain part of the controller.

Learned solution maps and approximate MPC. Learning-to-optimize methods map the parameters of a static problem instance to a near-optimal solution, including hard-constraint completion and correction (Donti et al., 2021) and single-period AC and DC optimal power flow proxies (Zamzam & Baker, 2020; Fioretto et al., 2020; Pan et al., 2021; Park & Van Hentenryck, 2023). Approximate MPC instead learns a receding-horizon control law offline and deploys it in closed loop, in some cases with guarantees (Chen et al., 2018b; Hertneck et al., 2018; Karg & Lucia, 2020). Pareto set learning represents a trade-off front with a preference-conditioned model (Navon et al., 2021; Lin et al., 2022). These methods learn a map for a fixed objective or an externally supplied preference. GPC shares their amortization premise but addresses closed-loop, time-coupled multi-objective control: priorities are recomputed from physical urgency at every cycle, ramp and inter-step limits couple successive decisions, the learned response is re-localized and locally corrected online, and the deployment analysis accounts for the final executed response across repeated cycles.

Offline and multi-objective learning. Offline RL and sequence-modeling approaches learn from stored trajectories or expert data (Kostrikov et al., 2022; Kumar et al., 2020; Fujimoto & Gu, 2021; Chen et al., 2021; Janner et al., 2021); fixed-data studies also emphasize representation bias and distribution shift (François-Lavet et al., 2019; Fujimoto et al., 2023). Multi-objective optimization and preference-conditioned RL represent trade-offs through Pareto sets, scalarizations, or preference vectors (Deb et al., 2002; Zhang & Li, 2007; Roijers et al., 2013; Van Moffaert & Nowé, 2014; Abels et al., 2019; Pirotta et al., 2015; Reymond et al., 2022; Michailidis et al., 2026). GPC differs by deriving its deployment priority from physical urgency and moving within a learned solution representation rather than selecting only among fixed policies or externally supplied preferences.

Model-based and physics-informed control. Model-based RL uses learned or probabilistic dynamics to improve planning and data efficiency (Ghahramani, 2015; Chua et al., 2018; Moerland et al., 2023). Physics-informed and structure-preserving models place known equations or invariants inside the learned representation (Raissi et al., 2019; Karniadakis et al., 2021; de Avila Belbute-Peres et al., 2018; Hu et al., 2020; Chen et al., 2018a; Greydanus et al., 2019; Brunton et al., 2016; Korda & Mezić, 2018). Classical nonlinear optimization, dynamic programming, and MPC instead solve directly against an explicit model (Garcia et al., 1989; Bertsekas, 2016; 2017). GPC occupies the space between these approaches: it learns from constrained reference responses and reuses their structure online, trading offline construction effort for a cheaper response update.

Geometric tools and physical applications. The local metric, retraction, and group-valued correction used by GPC draw on differential geometry, geometric mechanics, and Riemannian optimization (Lee, 2000; 2003; 2018; Bullo & Lewis, 2005; Murray et al., 2017; Absil et al., 2008; Boumal, 2023; Barfoot, 2024; De Bortoli et al., 2022; Huang et al., 2022). These tools describe local movement on a represented set; GPC couples them to physical urgency, constrained response

data, and repeated closed-loop re-localization. The motivating application regime includes learning-based power, transportation, and communication control (Su et al., 2025; Haydari & Yılmaz, 2022; Zong et al., 2025; Luong et al., 2019; Feriani & Hossain, 2021), where model structure and hard operating limits are commonly available.

C Complete Deployment, Geometry, and Integration Theory

The proofs below retain the central feasibility and response-error guarantee and provide the complete supporting chain: decoder regularity, finite-cover certification, localization, local integration, contraction, and per-cycle reset. They distinguish a generic geometric navigator from a domain implementation that uses a learned plan head and candidate refinement. In particular, an RK2 or Lie-group result applies only to an implementation that actually uses the stated update. The main tube theorem needs the total accepted displacement and final output map, not a particular integrator.

C.1 One-cycle deployment tube and proofs

Theorem C.1 (Conditional one-cycle deployment tube)**.** *Fix a control cycle t. Let $\mathcal{N}_r(\mathcal{M}_t) = \{z : \operatorname{dist}(z, \mathcal{M}_t) \le r\}$, where $r = \delta_{\mathrm{loc}} + \delta_{\mathrm{step}}$. Suppose the following hold:*

1. *The final map $A_t : \mathcal{N}_r(\mathcal{M}_t) \to \mathcal{Y}_t$ is K-Lipschitz. On an output region containing $A_t(\mathcal{N}_r(\mathcal{M}_t))$, each $f_{t,i}$ is L_i-Lipschitz and g_t is L_g-Lipschitz in the chosen residual norm.*
2. *At every $a \in \mathcal{M}_t$, $f_{t,i}(A_t(a)) \le -\gamma_i$ with $\gamma_i > 0$, $\|g_t(A_t(a))\| \le \eta_0$, and $\|A_t(a) - y_t^{\mathrm{ref}}(a)\| \le \delta_{\mathrm{dec}}$.*
3. *Localization and the* total accepted *update satisfy $\operatorname{dist}(z_t^0, \mathcal{M}_t) \le \delta_{\mathrm{loc}}$ and $\|z_t^+ - z_t^0\| \le \delta_{\mathrm{step}}$.*

Choose a nearest $a_t \in \mathcal{M}_t$ to z_t^+. Then

$$f_{t,i}(A_t(z_t^+)) \le -\gamma_i + L_i K r, \qquad \|g_t(A_t(z_t^+))\| \le \eta_0 + L_g K r, \tag{18}$$

$$\|A_t(z_t^+) - y_t^{\mathrm{ref}}(a_t)\| \le K r + \delta_{\mathrm{dec}}. \tag{19}$$

All norms in this section are the Euclidean norms or the explicitly chosen output/residual norms in the theorem. Fix one physical control cycle t. A nonempty closed subset $\mathcal{M}_t$ of finite-dimensional Euclidean space admits a nearest point to z_t^+: a minimizing sequence is bounded, has a convergent subsequence, and its limit is in the set. Let this nearest point be a_t.

For any nonempty set, its distance function is 1-Lipschitz. Indeed for every $a \in \mathcal{M}_t$, $\|z - a\| \le \|z - z'\| + \|z' - a\|$; taking infima and exchanging z, z' gives the claim. Hence

$$\|z_t^+ - a_t\| = \operatorname{dist}(z_t^+, \mathcal{M}_t) \le \|z_t^+ - z_t^0\| + \operatorname{dist}(z_t^0, \mathcal{M}_t) \le \delta_{\mathrm{step}} + \delta_{\mathrm{loc}} = r. \tag{20}$$

Both z_t^+ and a_t belong to the certified latent region $\mathcal{N}_r(\mathcal{M}_t)$, and their decoded responses belong to the certified output region. Therefore Lipschitz continuity of the *final* output map gives

$$\|A_t(z_t^+) - A_t(a_t)\| \le K\|z_t^+ - a_t\| \le Kr. \tag{21}$$

Apply the constraint Lipschitz assumptions separately to every inequality and the equality residual:

$$f_{t,i}(A_t(z_t^+)) \le f_{t,i}(A_t(a_t)) + L_i\|A_t(z_t^+) - A_t(a_t)\| \le -\gamma_i + L_i K r, \tag{22}$$

$$\|g_t(A_t(z_t^+))\| \le \|g_t(A_t(a_t))\| + \|g_t(A_t(z_t^+)) - g_t(A_t(a_t))\| \le \eta_0 + L_g K r. \tag{23}$$

For the calibrated matched reference,

$$\|A_t(z_t^+) - y_t^{\mathrm{ref}}(a_t)\| \le \|A_t(z_t^+) - A_t(a_t)\| + \|A_t(a_t) - y_t^{\mathrm{ref}}(a_t)\| \tag{24}$$

$$\le K r + \delta_{\mathrm{dec}}. \tag{25}$$

Equations (23)–(25) prove Theorem C.1. For Theorem 1, the constants are uniform over the cycles in question, so the same inequalities hold at every t without taking a steady-state limit. The conditions $L_i K r \le \gamma_i$ for every i and $\eta_0 + L_g K r \le \eta$ give inequality feasibility and the stated equality

tolerance throughout every finite rollout. This also covers a constant constraint or constant decoder when a Lipschitz constant is zero, without dividing by zero.

Finally, coordinate extraction to the first executed action has operator norm at most one, so applying Π_t to both the response and its matched reference preserves the same error upper bound at every cycle. Taking the supremum over $0 \le t < T$ proves the non-cumulative action-error claim in Theorem 1. No previous-cycle latent error appears because each cycle starts from a fresh localization of the current measurement.

C.2 Decoder Lipschitz constants and final output transformations

Proposition 2 (Composition and spectral-norm bounds)**.** *Suppose a feedforward decoder is a composition of affine layers with weight matrices $W_1, \dots, W_L$ and activations with Lipschitz constants $\ell_1, \dots, \ell_{L-1}$. Then a valid Euclidean Lipschitz constant is*

$$K_D \le \left(\prod_{j=1}^{L} \|W_j\|_2 \right) \left(\prod_{j=1}^{L-1} \ell_j \right).$$

If T is a K_T-Lipschitz final transformation, $A = T \circ D$ is $K_T K_D$-Lipschitz. Coordinatewise clipping to a fixed box has $K_T = 1$.

Proof. For an affine layer, the bias cancels in a difference and $\|Wz - Wz'\| \le \|W\|_2 \|z - z'\|$. Applying this inequality and the activation bounds successively through all layers yields the product bound. The composition claim follows from $\|T(D(z)) - T(D(z'))\| \le K_T \|D(z) - D(z')\|$. For scalar clipping, truncation to an interval can only reduce the distance between two real inputs. Square this inequality in each coordinate and sum to obtain the Euclidean box result. □

Spectral normalization controls an upper singular value. It does not ensure a positive lower singular value, injectivity, or a physical margin. A residual network instead has a bound obtained from the appropriate sums and products, for example $z \mapsto z + f(z)$ has constant at most $1 + K_f$. These are architecture-level bounds only when the layer norms are actually bounded as stipulated.

If the output includes both a decoded state and control, the product-norm Lipschitz constant can be bounded by $\sqrt{K_s^2 + K_u^2}$. Subsequent refinement, projection, or state correction must be included in T or bounded separately. A discontinuous candidate-selection map does not automatically have a uniform Lipschitz constant; the theorem can be applied to each verified accepted candidate and its actual displacement, or with a separately bounded output perturbation.

Proposition 3 (A separately bounded output perturbation)**.** *Suppose a base map A satisfies the tube estimates, and a final response y^+ obeys $\|y^+ - A(z^+)\| \le \delta_{\text{out}}$. Then*

$$f_i(y^+) \le -\gamma_i + L_i(Kr + \delta_{\text{out}}), \tag{26}$$

$$\|g(y^+)\| \le \eta_0 + L_g(Kr + \delta_{\text{out}}), \tag{27}$$

$$\|y^+ - y^{\text{ref}}(a)\| \le Kr + \delta_{\text{dec}} + \delta_{\text{out}}. \tag{28}$$

Proof. Apply each function's Lipschitz bound between y^+ and $A(z^+)$, then use the corresponding tube bound. The response inequality is the triangle inequality. □

Clipping enforces its own box but can violate another constraint. For example, clipping $(2, -1)$ to $[0, 1]^2$ changes $(2, -1)$ to $(1, 0)$; a coupled constraint such as $u_1 + 2u_2 = 0$, satisfied by the first point, is not satisfied by the clipped point. Thus certificates must be evaluated after all relevant transformations.

C.3 Finite-cover certification, including changing contexts

Proposition 4 (Verified anchors imply a uniform certificate)**.** *Let $\{a_j\}_{j=1}^N \subseteq \mathcal{M}_t$ be a finite q_{cov}-cover of $\mathcal{M}_t$: every $a \in \mathcal{M}_t$ is within q_{cov} of an anchor. Suppose verified anchor responses satisfy*

$$f_{t,i}(A_t(a_j)) \le -\widehat{\gamma}_i, \qquad \|g_t(A_t(a_j))\| \le \widehat{\eta}_0, \qquad \|A_t(a_j) - y_t^{\text{ref}}(a_j)\| \le \widehat{\delta}.$$

If the map and constraint constants are as in the main theorem, and y_t^{ref} is K_{ref}-Lipschitz, then uniform constants are

$$\gamma_i = \widehat{\gamma}_i - L_i K q_{\mathrm{cov}}, \quad \eta_0 = \widehat{\eta}_0 + L_g K q_{\mathrm{cov}}, \quad \delta_{\mathrm{dec}} = \widehat{\delta} + (K + K_{\mathrm{ref}}) q_{\mathrm{cov}}. \tag{29}$$

Proof. For any a choose a_j with $\|a - a_j\| \le q_{\mathrm{cov}}$. Then $f_{t,i}(A_t(a)) \le f_{t,i}(A_t(a_j)) + L_i K q_{\mathrm{cov}} \le -\widehat{\gamma}_i + L_i K q_{\mathrm{cov}}$. For the equality residual, the triangle inequality gives the second constant. For calibration insert both anchor values:

$$\|A_t(a) - y_t^{\mathrm{ref}}(a)\| \le \|A_t(a) - A_t(a_j)\| + \|A_t(a_j) - y_t^{\mathrm{ref}}(a_j)\| + \|y_t^{\mathrm{ref}}(a_j) - y_t^{\mathrm{ref}}(a)\|.$$

Bounding the three terms proves the last constant. □

Only a positive resulting γ_i certifies a nonzero inequality tube. A finite set of training observations does not establish its own covering radius.

For contextual certification, define $H_i(c, z) = f_i(c, A(c, z))$. Suppose it is Lipschitz separately in latent and context coordinates:

$$|H_i(c, z) - H_i(c', z')| \le L_i^z \|z - z'\| + L_i^c d_c(c, c').$$

If every current pair (c, z) is within (q_z, q_c) of a verified pair, then $H_i(c, z) \le -\widehat{\gamma}_i + L_i^z q_z + L_i^c q_c$, by one application of this inequality. Equality and calibration residuals obey the same construction with their own constants. The previous executed control must be in c for ramp constraints; otherwise an old-context certificate need not concern the current feasible set. Random or Latin hypercube sampling supplies a sample design, not a deterministic joint covering certificate.

C.4 Equalities, exact dynamics, and model error

A strict inequality margin cannot establish exact equality. If an equality is encoded as $g(y) \le 0$ and $-g(y) \le 0$, both sides cannot simultaneously have a positive slack at a point satisfying $g(y) = 0$. For exact feasibility one must eliminate the equality by a valid parameterization, propagate exact modeled dynamics, or apply an exact correction. When the decoder predicts voltage and dispatch together, its output need not lie on the AC balance manifold; a numerical tolerance must be specified.

Proposition 5 (Constraint-model discrepancy)**.** *Suppose, throughout the certified output region, $f_i^{\mathrm{true}}(y) \le f_i^{\mathrm{model}}(y) + \beta_i$ and $\|g^{\mathrm{true}}(y) - g^{\mathrm{model}}(y)\| \le \beta_g$. Then the tube bounds imply*

$$f_i^{\mathrm{true}}(A(z^+)) \le -\gamma_i + L_i K r + \beta_i, \qquad \|g^{\mathrm{true}}(A(z^+))\| \le \eta_0 + L_g K r + \beta_g.$$

Proof. Substitute the modeled constraint bounds into the assumed discrepancy inequality; use the triangle inequality for g. □

These are properties of the evaluated final response. A predicted auxiliary state represents the physical state only under the corresponding model/measurement assumptions. A controller cannot establish actual future-state safety merely by decoding an internally consistent state-control pair.

C.5 Localization: what observation residuals can establish

Let $\mathcal{O}(z)$ denote the decoded observation map, such as $h_{\mathrm{obs}}(D_s(z))$. The main theorem uses a verified latent localization distance. The following sufficient local condition explains how observation reconstruction may contribute.

Proposition 6 (A local inverse bound with a verified chart)**.** *Let $\mathcal{O}$ be continuously differentiable on a convex ball around a. Assume $J_{\mathcal{O}}(a)$ has full column rank. Suppose $\sigma_{\min}(J_{\mathcal{O}}(a)) \ge \sigma_0 > 0$ and $\sup_z \|J_{\mathcal{O}}(z) - J_{\mathcal{O}}(a)\|_2 \le \sigma_0/2$ on that ball. For all z, z' in the ball,*

$$\|\mathcal{O}(z) - \mathcal{O}(z')\| \ge \tfrac{\sigma_0}{2} \|z - z'\|.$$

If $x = \mathcal{O}(a) + \nu$, $\|\nu\| \le \delta_x$, and $\|\mathcal{O}(z) - x\| \le \delta_{\mathcal{O}}$, then $\|z - a\| \le 2(\delta_{\mathcal{O}} + \delta_x)/\sigma_0$.

Proof. Integrating the Jacobian along the line from z' to z gives

$$\mathcal{O}(z) - \mathcal{O}(z') = J_{\mathcal{O}}(a)(z - z') + \int_0^1 [J_{\mathcal{O}}(z' + \tau(z - z')) - J_{\mathcal{O}}(a)](z - z')\,\mathrm{d}\tau.$$

The first term has norm at least $\sigma_0\|z - z'\|$, and the second has norm at most $\sigma_0\|z - z'\|/2$. The reverse triangle inequality proves the first claim. Set $z' = a$, then use $\|\mathcal{O}(z) - \mathcal{O}(a)\| \le \|\mathcal{O}(z) - x\| + \|\nu\|$ to obtain the second. □

This result is local and branch-specific. A folded map can have separated latent points with the same observation, and a pointwise full-rank Jacobian does not give a global inverse bound. Distinct preferences may share the same physical observation while requiring different actions. Reconstruction and locality losses encourage a useful map but do not certify this chart condition.

If localization uses a smoothed update $z^0 = (1-\alpha_{\mathrm{loc}})z^- + \alpha_{\mathrm{loc}}\hat{a}$ with $\hat{a} \in \mathcal{M}_t$, then $\mathrm{dist}(z^0, \mathcal{M}_t) \le (1 - \alpha_{\mathrm{loc}})\|z^- - \hat{a}\|$, because $\hat{a}$ is an admissible comparison point. The blend need not lie in a nonconvex $\mathcal{M}_t$. If the observation-consistent set is empty, an exact projection to it does not exist; a fallback or a new verified residual bound is needed.

C.6 Restoration potentials and valid contraction recurrences

Let $\Psi(z) \ge 0$ be a differentiable restoration potential with minimum zero. Its relation to geometric distance is a separate error-bound assumption, for example

$$\mathrm{dist}(z, \mathcal{M})^2 \le C_\Psi \Psi(z). \tag{30}$$

This is not implied by a generic reconstruction loss. Consider a region invariant under the updates below.

Proposition 7 (Continuous and discrete restoration)**.** *Suppose $\|\nabla\Psi(z)\|^2 \ge 2\mu\Psi(z)$ with $\mu > 0$. The continuous snap flow $\dot{z} = -\epsilon_{\mathrm{snap}}^{-1}\nabla\Psi(z)$ satisfies*

$$\Psi(z(t)) \le e^{-2\mu t/\epsilon_{\mathrm{snap}}}\Psi(z(0)).$$

If Ψ is also L_Ψ-smooth along each discrete update segment, then for $z^+ = z - \eta_\Psi\nabla\Psi(z)$, $0 < \eta_\Psi \le 1/L_\Psi$ and $0 < \mu\eta_\Psi < 1$,

$$\Psi(z^+) \le (1 - \mu\eta_\Psi)\Psi(z).$$

Proof. Along the continuous flow, $\dot{\Psi} = -\epsilon_{\mathrm{snap}}^{-1}\|\nabla\Psi\|^2 \le -2\mu\Psi/\epsilon_{\mathrm{snap}}$. Integrate this scalar inequality to obtain the first result. For the discrete update, smoothness gives

$$\Psi(z - \eta_\Psi\nabla\Psi) \le \Psi(z) - \eta_\Psi(1 - L_\Psi\eta_\Psi/2)\|\nabla\Psi\|^2.$$

As $1 - L_\Psi\eta_\Psi/2 \ge 1/2$, the PL inequality implies $\Psi(z^+) \le \Psi(z) - \mu\eta_\Psi\Psi(z)$. □

The factor $1 - L\eta/2$ cannot be discarded. An explicit snap with $\eta_\Psi = h/\epsilon_{\mathrm{snap}}$ requires a small enough ratio $h/\epsilon_{\mathrm{snap}}$; a large-timescale-separation limit with $h/\epsilon_{\mathrm{snap}} \gg 1$ does not establish its discrete stability. Exact snap flow or a separately analyzed stable implicit correction is a different implementation.

For a least-squares residual $\Psi(z) = \|\mathcal{O}(z) - x\|^2/2$, one has $\nabla\Psi = J_{\mathcal{O}}^\top(\mathcal{O}(z) - x)$. A lower bound on the smallest singular value controls this product only for residual components in the range of $J_{\mathcal{O}}$; perpendicular residual components are annihilated. Thus PL and geometric error bounds should be assumed or verified separately unless a precise residual-range argument is available.

Proposition 8 (Forced recurrence and non-accumulation)**.** *Suppose accepted updates obey $\Psi_{k+1} \le q_{\mathrm{ctr}}\Psi_k + b_k$ with $0 \le q_{\mathrm{ctr}} < 1$ and $0 \le b_k \le b$. Then*

$$\Psi_k \le q_{\mathrm{ctr}}^k\Psi_0 + \sum_{j=0}^{k-1} q_{\mathrm{ctr}}^{k-1-j}b_j \le q_{\mathrm{ctr}}^k\Psi_0 + \frac{b(1 - q_{\mathrm{ctr}}^k)}{1 - q_{\mathrm{ctr}}}. \tag{31}$$

Hence $\limsup_k \Psi_k \le b/(1 - q_{\mathrm{ctr}})$ and $\sup_k \Psi_k \le \max\{\Psi_0, b/(1 - q_{\mathrm{ctr}})\}$. Under (30)*, multiply these bounds by C_Ψ and take square roots to bound distance.*

Proof. Substitute the recurrence into itself k times; induction gives the first inequality. The finite geometric sum gives the second. The right side is a convex combination of Ψ_0 and $b/(1-q_{\mathrm{ctr}})$, giving the uniform bound, and $q_{\mathrm{ctr}}^k \to 0$ gives the limit superior. □

A recurrence supplies an upper bound; it does not establish that the actual sequence converges to the fixed point. If the context changes, the resulting potential jump must be included in b_k. If such changes have no uniform bound, the fixed-context recurrence does not give a uniform tracking result.

Proposition 9 (Bounding the forcing from an actual perturbation)**.** *If $\|\nabla\Psi\| \le B_\Psi$ throughout a perturbation segment and $\|\delta_k\| \le d_k$, then $\Psi(z+\delta_k) - \Psi(z) \le B_\Psi d_k$. If Ψ is L_Ψ-smooth and the gradient is known at z, then the sharper estimate is*

$$\Psi(z+\delta_k) - \Psi(z) \le \|\nabla\Psi(z)\|d_k + \tfrac{L_\Psi}{2}d_k^2.$$

Proof. Integrate $\nabla\Psi(z+\tau\delta_k)^\top\delta_k$ over $\tau \in [0,1]$ for the first claim. For the second subtract $\nabla\Psi(z)$ inside that integral and use $L_\Psi\tau\|\delta_k\|$ as the gradient-difference bound. □

Thus an $O(h^2)$ latent perturbation generally produces an $O(h^2)$ potential perturbation in a fixed tube, with an $O(h^4)$ term at a zero-gradient base point. A special expression such as $b = \epsilon_{\mathrm{snap}}V_{\max}^2h^2/2$ cannot be selected freely; it must follow from verified constants and the actual correction.

C.7 Midpoint accuracy and local geometric corrections

Proposition 10 (Euler and midpoint local error)**.** *Let $\mathcal{F}$ be twice continuously differentiable with bounded derivatives on a region containing the exact and numerical updates. With a fixed context and step h, Euler has local error $O(h^2)$ and midpoint RK2 has local error $O(h^3)$:*

$$z_{\mathrm{E}} = z + h\mathcal{F}(z), \qquad z_{\mathrm{M}} = z + h\mathcal{F}(z + \tfrac{h}{2}\mathcal{F}(z)).$$

On a finite interval where both methods stay in this region and $\mathcal{F}$ is Lipschitz, the respective global errors are $O(h)$ and $O(h^2)$.

Proof. The exact ODE solution has

$$z(h) = z + h\mathcal{F}(z) + \tfrac{h^2}{2}J_{\mathcal{F}}(z)\mathcal{F}(z) + O(h^3).$$

Euler omits the quadratic term. Taylor expansion of the midpoint field gives $\mathcal{F}(z + h\mathcal{F}(z)/2) = \mathcal{F}(z) + (h/2)J_{\mathcal{F}}(z)\mathcal{F}(z) + O(h^2)$, so midpoint includes it and differs only by $O(h^3)$. For a one-step map with Lipschitz factor at most $1+Ch$ and local error at most C_0h^{p+1}, the global error obeys $e_{k+1} \le (1+Ch)e_k + C_0h^{p+1}$. Iteration over $kh \le T$ gives $e_k \le e^{CT}e_0 + C_0h^p(e^{CT}-1)/C$ when $C > 0$, and the analogous C_0Th^p bound when $C = 0$. Take $p = 1$ or 2. □

The same statement applies to a capped field only where the capped field has the required smoothness. At the cap transition it may be merely Lipschitz. A retraction or Lie residual must have an appropriate local-error estimate to preserve second order; adding an arbitrary $O(h^2)$ correction can alter the quadratic coefficient and reduce the order. The tube theorem remains valid using actual displacement.

Proposition 11 (Matrix exponential and rotation residual bounds)**.** *For any matrix X with a submultiplicative norm,*

$$\|\exp X - I - X\| \le \tfrac{1}{2}\|X\|^2 e^{\|X\|}.$$

For a skew-symmetric X in the Euclidean operator norm, $\|\exp X\|_2 = 1$ and $\|\exp X - I\|_2 \le \|X\|_2$. If B has orthonormal columns, $d = B^\top\Delta z$, $R = \exp X$ is a blockwise rotation, and $\Delta z_L = B(R-I)d$, then

$$\|\Delta z_L\| \le \|X\|_2\|\Delta z\|.$$

In particular, if $\|\Delta z\| \le hV$ and $\|X\|_2 \le hC_\omega V$, then $\|\Delta z_L\| \le C_\omega V^2h^2$.

Proof. The series remainder is bounded by $\sum_{j\ge2}\|X\|^j/j!$. For $j \ge 2$, $j! \ge 2(j-2)!$, giving the first bound. For skew-symmetric X, $(e^X)^\top e^X = e^{-X}e^X = I$, proving orthogonality. The integral identity $e^X - I = \int_0^1 e^{sX}X\,\mathrm{d}s$ then gives $\|e^X - I\|_2 \le \|X\|_2$. Finally $\|B\|_2 = \|B^\top\|_2 = 1$, so $\|B(R-I)B^\top\Delta z\| \le \|X\|_2\|\Delta z\|$, and substitute the stated bounds. □

For an incomplete block implemented by zero-padding to three dimensions and then truncating, let E be the isometric embedding and $Q = E^\top$ the coordinate extraction. Its map is QRE, which is nonexpansive but need not preserve norm. The residual satisfies $\|QRE - I\|_2 = \|Q(R-I)E\|_2 \le \|R - I\|_2 \le \|X\|_2$, so the residual estimate remains valid. Exact norm preservation applies only to a full rotation block.

A decoder metric $G = J_{D_s}^\top J_{D_s} + \lambda_m I$ has eigenvalues at least $\lambda_m > 0$. Thus a local angular parameter derived by dividing projected velocity by square-root eigenvalues can have a bound of order $V/\sqrt{\lambda_m}$, multiplied by any configured rotation scale. These constants must be retained in C_ω. The exponential remainder does not have coefficient $1/2$ uniformly after discarding $e^{\|X\|}$.

If the accepted update is $\mathrm{Retr}_z(\Delta z + \gamma_L \Delta z_L)$ and $\|\mathrm{Retr}_z(v) - z\| \le C_R\|v\|$ on its domain, then

$$\|z^+ - z\| \le C_R(hV + \gamma_L C_\omega V^2 h^2).$$

This follows by the triangle inequality. Any candidate refinement after this update requires another displacement bound or direct measurement of the final displacement.

For Proposition 1 in the main text, let $z^{\mathrm{geom}} = \mathrm{Retr}_z(\Delta z + \gamma_L \Delta z_L)$ and suppose a later accepted refinement obeys $\|z^+ - z^{\mathrm{geom}}\| \le \delta_{\mathrm{refine}}$. The bound follows directly from (C.7) and $\|z^+ - z\| \le \|z^+ - z^{\mathrm{geom}}\| + \|z^{\mathrm{geom}} - z\|$.

C.8 Ambient Lie closure and physical feasibility

If the latent coordinates belong to a Lie group $\mathcal{G}$, closure ensures $z\exp(hv) \in \mathcal{G}$ for a group-compatible tangent vector. It does not imply membership in a learned subset $\mathcal{M} \subset \mathcal{G}$, nor does it imply decoded physical feasibility. These are different sets.

Proposition 12 (Feasibility following a group or chart update)**.** *Suppose the final map and reference set satisfy the main tube assumptions. Let any accepted group, chart, retraction, or candidate update obey $\|z^+ - z^0\| \le \delta_{\mathrm{step}}$ in the certified chart norm. If $L_i K(\delta_{\mathrm{loc}} + \delta_{\mathrm{step}}) \le \gamma_i$ for every inequality and $\eta_0 + L_g K(\delta_{\mathrm{loc}} + \delta_{\mathrm{step}}) \le \eta$, then the final response satisfies those inequalities and equality tolerance.*

Proof. The update type is immaterial to (20); it uses only the accepted displacement and localization distance. Apply (23). □

This proof does not require convexity of the feasible set or scalarized objective. Positive reach or a unique nearest-point projection is also unnecessary for this main feasibility argument: some nearest point suffices. Reach becomes relevant only for differentiating distance or proving a second-order tangency estimate.

C.9 Tangential updates and second-order off-set error

The following result states the additional assumptions needed for the stronger second-order geometric claim.

Proposition 13 (A sufficient second-order departure bound)**.** *Let $a \in \mathcal{M}$ lie on a smooth embedded manifold. Suppose a comparison curve $\gamma : [0, h] \to \mathcal{M}$ satisfies $\gamma(0) = a$, $\dot\gamma(0) = v$, and $\|\ddot\gamma(s)\| \le B_{\mathcal{M}}$. Suppose the implemented ambient update curve c has $c(0) = a$, $\dot c(0) = v$, and $\|\ddot c(s)\| \le B_{\mathrm{int}}$. Then*

$$\mathrm{dist}(c(h), \mathcal{M}) \le \tfrac{1}{2}(B_{\mathrm{int}} + B_{\mathcal{M}})h^2.$$

Proof. Taylor's formula with integral remainder gives $c(h) = a + hv + \int_0^h (h-s)\ddot c(s)\,\mathrm{d}s$ and the analogous expression for $\gamma(h)$. Subtract and bound: $\|c(h) - \gamma(h)\| \le (B_{\mathrm{int}} + B_{\mathcal{M}})h^2/2$. Since $\gamma(h) \in \mathcal{M}$, the distance to $\mathcal{M}$ cannot exceed this norm. □

For a straight Euler tangent step $c(s) = a + sv$, $B_{\mathrm{int}} = 0$. For a constant-speed intrinsic geodesic and a bounded second fundamental form, $B_{\mathcal{M}} \le \kappa\|v\|^2$ on the curve. An ambient group exponential generally has its own acceleration B_{int}; manifold curvature alone does not bound it. Tangency is essential: a nonzero initial normal velocity produces a first-order $O(h)$ departure. The proposition

also requires the comparison curve and acceleration bounds throughout the step, not just a pointwise curvature value.

C.10 Per-cycle reset and uniform action error

Proposition 14 (Uniform matched-reference error after approximate reset)**.** *Suppose each measured context supplies a reference set $\mathcal{M}_t$, localization satisfies $\operatorname{dist}(z_t^0, \mathcal{M}_t) \le \delta_{\mathrm{loc}}$, and every accepted update moves at most δ_{step}. Suppose the final decoder is K-Lipschitz and calibrated on $\mathcal{M}_t$ to error δ_{dec}, uniformly in t. Then*

$$\sup_t \|A_t(z_t^+) - y_t^{\mathrm{ref}}(a_t)\| \le K(\delta_{\mathrm{loc}} + \delta_{\mathrm{step}}) + \delta_{\mathrm{dec}},$$

where a_t is a nearest current-set point to z_t^+. If the reset is exact and the hypotheses of Proposition 13 hold at every cycle with common constants, one may instead use $s_\perp = (B_{\mathrm{int}} + B_{\mathcal{M}})h^2/2$ as the off-set distance bound, giving $Ks_\perp + \delta_{\mathrm{dec}}$.

Proof. The first claim is (20) and (25) applied anew each cycle. For the second, the exact reset starts at $a \in \mathcal{M}_t$; Proposition 13 bounds distance of the final point directly by $s_\perp$. Choose its nearest reference and apply decoder Lipschitz continuity and calibration. Uniform constants give the supremum. □

An approximate reset does not automatically inherit the exact-reset second-order estimate. It can do so with an additional perturbation-stability bound for the entire update, but the simpler main theorem already handles it using total displacement. No exact reset or tangency condition is inferred from a generic learned initializer.

A different useful guarantee follows from a verified energy error bound (30) and the forced recurrence:

$$\limsup_k \|A(z_k) - y^{\mathrm{ref}}(a_k)\| \le K\sqrt{\frac{C_\Psi b}{1 - q_{\mathrm{ctr}}}} + \delta_{\mathrm{dec}}.$$

Indeed (31) bounds the energy, (30) converts it to distance, and decoder calibration gives the action bound. A uniform version includes $\sqrt{C_\Psi \max\{\Psi_0, b/(1-q_{\mathrm{ctr}})\}}$. This preserves the initialization term and contraction denominator; a limit-superior statement is not automatically a uniform bound with the same right side.

C.11 Matched responses, priorities, and physical trajectory errors

Proposition 15 (The additional response-selection term)**.** *Let $y_t^\star$ be a desired current-context reference, for example an exact solution for a specified priority. Then*

$$\|A_t(z_t^+) - y_t^\star\| \le Kr + \delta_{\mathrm{dec}} + \|y_t^{\mathrm{ref}}(a_t) - y_t^\star\|, \tag{32}$$

$$|C_t(A_t(z_t^+)) - C_t(y_t^{\mathrm{ref}}(a_t))| \le L_C(Kr + \delta_{\mathrm{dec}}) \tag{33}$$

for any L_C-Lipschitz scalar cost on the compared outputs.

Proof. Insert the matched reference into the first norm and apply the triangle inequality and main action bound. The second is cost Lipschitz continuity composed with the same bound. □

The response-selection term includes finite offline-solver accuracy, limited representation coverage, and online movement along the response family. Off-set distance alone does not control it.

For context, a global minimizer of a positive-weight scalarization is Pareto optimal: if another feasible point weakly improved every objective and strictly improved one, its positive-weight sum would be strictly smaller, contradicting optimality. A local nonlinear solver does not establish this global premise. Moreover, no positive weight can select an unsupported Pareto point. These facts describe the target construction and do not add a global guarantee to the tube theorem.

The difference between action consistency and state stability can be stated precisely. Suppose a true closed loop has a uniformly contractive ideal map $T_t(x) = F_t^{\mathrm{phys}}(x, \pi_t^\star(x))$ with $\|T_t(x) -$

$T_t(x')\| \le a\|x - x'\|$, $a < 1$. Suppose F_t^{phys} is L_u-Lipschitz in action and the implemented policy has $\|\pi_t(x) - \pi_t^\star(x)\| \le e_u$ throughout the relevant state region. For trajectories of the two policies,

$$\|x_{t+1} - x_{t+1}^\star\| \le a\|x_t - x_t^\star\| + L_u e_u.$$

To prove this, insert $F_t^{\text{phys}}(x_t, \pi_t^\star(x_t))$ between the two next states and apply action Lipschitz continuity and contraction. Iterating gives

$$\|x_t - x_t^\star\| \le a^t\|x_0 - x_0^\star\| + \frac{L_u e_u(1 - a^t)}{1 - a}.$$

This is an additional dynamics result with explicit contraction assumptions. The geometric theorem alone supplies neither $a < 1$ nor the desired-policy action bound e_u without its response-selection term.

C.12 Connection to rollout-aware decoder supervision

The distribution-shift, paired-target coupling, weighted-risk, and uniform supervision proofs are provided in Appendix G. Their role is to control how offline interpolation and geometry-corrected training latents cover the post-update queries of a frozen decoder. They complement the pointwise calibration premise here but do not replace a uniform certificate by an average loss. Synthetic oracle-transition latents are not automatically equivalent to states produced by recursively executing the learned controller.

If all encoders, the state decoder, and the geometric update are frozen during a decoder-only stage, a penalty depending only on those frozen latent quantities has zero gradient with respect to decoder parameters. Such terms can be recorded as diagnostics, or become trainable regularizers only in a stage that updates their dependency graph. This follows directly from differentiating a parameter-independent quantity.

D Extended Offline Pareto-Map Construction

D.1 Context Sampling and Complete Reference Records

The object represented by GPC is a family of constrained responses indexed by operating context and trade-off, rather than a static Pareto curve in action space. The context includes state, exogenous parameters, and the previous action. When a response spans a horizon, the reference solver propagates the dynamics and evaluates all intermediate constraints. The observation can contain only the components available at deployment, whereas the offline record can also contain the physical state and reference objective values.

For each sampled context $c^{(i)}$, choose a set of weights $\mathcal{W} \subset \Delta^{n_J - 1}$ and solve the constrained problem for each weight. The retained record has the form

$$(x^{(i)}, s^{(i)}, p^{(i)}, u_{-1}^{(i)}, w^{(r)}, \sigma^{(i,r)}, U^{\text{ref,(i,r)}}, \mathbf{J}^{(i,r)}, \boldsymbol{\gamma}^{(i,r)}). \tag{34}$$

Here $\boldsymbol{\gamma}$ contains separate physical margins, rather than a single aggregate feasibility score. For robot navigation these include obstacle, inter-robot, and control-slew margins. For OPF they include generator, thermal, voltage, and ramp margins, with equality residuals recorded separately. A successful numerical solve can still lie on an active inequality boundary and have zero margin. Strict-margin certification therefore requires an additional verified slack or a smaller certified subset.

Latin hypercube sampling. For a bounded parameter coordinate $p_j \in [a_j, b_j]$, partition its marginal range into N_c equal-probability strata. With a random permutation π_j of $\{0, \dots, N_c - 1\}$ and independent $\xi_{ij} \in [0, 1)$, a uniform-marginal design is

$$p_{ij} = a_j + (b_j - a_j)\frac{\pi_j(i) + \xi_{ij}}{N_c}. \tag{35}$$

Each marginal is stratified, which reduces marginal clustering relative to unconstrained independent draws. This does not ensure a small covering radius in a high-dimensional joint parameter space. GPC's computational advantage depends on whether the response family over the sampled envelope is compressible; it does not remove the cost of resolving qualitatively different trajectory branches.

D.2 Scalarized Solutions and Their Coverage

For fixed context, the ideal single-stage scalarization is

$$U^\star(w;c) \in \arg\min_{U\in\mathcal{U}_{\text{feas}}(c)} w^\top \mathbf{J}(c,U), \qquad \mathcal{U}_{\text{feas}}(c) = \{U : g_c(U) = 0,\ f_c(U) \le 0\}. \tag{36}$$

A global minimizer with strictly positive weights is Pareto efficient: if another feasible response weakly improved every objective and strictly improved one, its weighted cost would be strictly lower. Weighted sums select supported efficient responses. Non-supported sections of a nonconvex objective front require other sampling schemes, such as ε-constraints or augmented Chebyshev scalarization, if their coverage is a design requirement. This study uses numerical weighted-sum references.

Numerical accuracy and multi-objective coverage are separate issues. A converged nonconvex IPOPT solve need not be a global minimizer, and a finite candidate sweep need not identify every feasible branch. We therefore use the notation U^{ref} for numerical targets. Representation training does not improve their optimality merely by embedding them. The OPF solve tolerance reported in the construction is 10^{-6}; it is distinct from the residual thresholds used to assess the final controller.

D.3 Physical-Urgency Feedback Labels

The urgency vector is a state-dependent feedback signal that encodes the current operating priority. Let $\delta_i(s,p) \in [0,1]$ be a normalized indicator for objective or recovery channel i. Define

$$\phi_i(s,p) = \exp\left(\frac{k_i\delta_i(s,p)}{\epsilon_i}\right) - 1, \qquad \sigma_i(s,p) = \frac{\phi_i(s,p)+\rho}{\sum_j(\phi_j(s,p)+\rho)}, \quad k_i \ge 0,\ \epsilon_i > 0,\ \rho > 0. \tag{37}$$

The offset keeps the denominator nonzero. If every indicator vanishes, the weights are equal. Economic or goal-directed dominance during nominal operation requires the corresponding background indicator and scales to dominate the other channels; it is not a consequence of the normalization alone. All labels in the offline dataset are computed using the same channel definitions that the corresponding navigator uses online.

In OPF, the thermal indicator is normalized worst branch loading, the voltage indicator is normalized worst voltage deviation, and a small constant economic indicator supplies a persistent background drive. Near a voltage or thermal boundary, the exponential map increases the corresponding weight continuously. In navigation, geometric clearance and goal-progress information play analogous roles. Indicators derived from a reconstructed state inherit that reconstruction's error. The feedback label is distinct from the scalarization weight that generated the reference: the latter explores the solution family, whereas the former closes the loop from the reconstructed state to operational priority.

When a stored transition includes the next true state s_{h+1}, it may also include σ_{h+1}. This optional label supports supervision of priority evolution. It does not make an observation-only encoder able to recover every possible weighted reference response for the same observation. Such ambiguity must be resolved by a conditioned input, a designated training anchor, or the subsequent priority-driven response update.

D.4 Dual-Path Losses and Their Roles

Write $z_k^x = E_x(x_k)$ and $z_k^o = E_o(s_k, \sigma_k, \mathbf{J}_k)$. The shared state decoder and response decoder support the following losses:

$$\mathcal{L}_{s,x} = \frac{1}{N}\sum_k \|s_k - D_s(z_k^x)\|^2, \qquad \mathcal{L}_{s,o} = \frac{1}{N}\sum_k \|s_k - D_s(z_k^o)\|^2, \tag{38}$$

$$\mathcal{L}_U = \frac{1}{N}\sum_k \|U_k^{\text{ref}} - D_U(z_k^x)\|^2. \tag{39}$$

State reconstruction forces each path to retain physical information. The action loss trains the deployable observation path to produce its assigned reference response. It is not mathematically automatic that the oracle path reconstructs an action from state and objectives; optional action supervision on that path can also be used, but is not part of the loss definition above.

Cross-modal consistency aligns direction and magnitude:

$$\mathcal{L}_c = \frac{1}{N}\sum_k \left[1 - \frac{\langle z_k^x, z_k^o\rangle}{\max(\|z_k^x\|\|z_k^o\|, \varepsilon_z)} + \beta\|z_k^x - z_k^o\|^2\right]. \tag{40}$$

The stabilizer $\varepsilon_z > 0$ avoids a singular denominator at zero codes. Alignment makes it possible to discard E_o online while retaining the reference encoder's organization in E_x.

Neighborhood regularization is defined over sampled operating conditions:

$$\mathcal{L}_n = \frac{1}{N}\sum_k \sum_{j\in\mathcal{N}_\kappa(k)} \exp\left(-\frac{\|p_k - p_j\|^2}{\sigma_p^2}\right)\|z_k^x - z_j^x\|^2. \tag{41}$$

This term favors local continuity; reconstruction prevents an entirely collapsed code from trivially minimizing it. A smooth latent representation is useful for action continuity, but neither this penalty nor an average reconstruction error guarantees a ramp limit, injectivity, or a uniform decoder bound. Those properties enter separately as assumptions or verified quantities in the theoretical analysis.

The complete representation objective weights action, both state paths, alignment, and neighborhood terms. The OPF training configuration uses three-layer MLPs with width 256, ReLU activations, spectral normalization, 500 Adam epochs, learning rate 10^{-3} with cosine annealing, alignment/locality weights $1.0/0.5$, and neighborhood size 10.

E Extended Online Localization and Geometric Navigation

E.1 Measurement-Consistent Localization and Reset

At time t, the current observation is encoded as $z_{\mathrm{obs,t}} = E_x(x_t)$. Noise, representation error, and operating shifts can put this code away from the desired reference region. Let x_t^{obs} be the measured-state component and h_{obs} its measurement map. The context-specific represented set is

$$\mathcal{M}_t = \left\{z \in \mathcal{M}^* : \|x_t^{\mathrm{obs}} - h_{\mathrm{obs}}(D_s(z))\|^2 \le \hat{\nu}_t + \tau_{\mathrm{geom}}\right\}. \tag{42}$$

The squared-noise allowance $\hat{\nu}_t$ and squared reconstruction-residual allowance τ_{geom} have different roles. The latter is calibrated from held-out residuals after training. An empirical calibration may make localization useful on typical observations, but does not guarantee that $\mathcal{M}_t$ is nonempty for every observation. If no consistent represented point is available, direct encoder initialization can still produce a candidate response, while a uniform localization certificate is unavailable for that cycle.

The localization design smooths the current nearest consistent code with the previous code:

$$z_t^0 = (1-\alpha_{\mathrm{loc}})z_{t-1}^+ + \alpha_{\mathrm{loc}}\,\mathrm{proj}_{\mathcal{M}_t}(z_{\mathrm{obs,t}}), \qquad \alpha_{\mathrm{loc}} \in (0,1]. \tag{43}$$

This reduces sensor-driven chattering and makes the current measurement enter every decision. It does not preserve membership in a nonconvex consistency set; smoothing and approximate projection must both be included in the measured localization error. Direct reset uses $z_t^0 = z_{\mathrm{obs,t}}$. The per-cycle action theorem needs a uniform bound on the actual localization, not simply the assertion that a reset exists.

A projection onto an implicitly represented nonlinear manifold can itself be computationally expensive. If localization is implemented with a finite set or a local approximation, its residual and search cost should be recorded. The method therefore distinguishes the abstract projection operator from a particular numerical localizer, and includes that implementation's cost in online timing.

E.2 Energy, Priority Filtering, and Intended Timescales

The reconstructed state $\hat{s}_t = D_s(z_t^0)$ determines urgency, with $\phi_{t,i} = \phi_i(\hat{s}_t, p_t)$. Optional first-order filtering uses

$$\sigma_t = (1-\alpha_\sigma)\sigma_{t-1} + \alpha_\sigma\bar{\sigma}_t, \qquad \bar{\sigma}_{t,i} = \frac{\phi_{t,i}+\rho}{\sum_j(\phi_{t,j}+\rho)}, \quad \alpha_\sigma \in (0,1]. \tag{44}$$

The energy combines restoration and task improvement:

$$\Phi_{\text{meas}}(z) = \|x_t^{\text{obs}} - h_{\text{obs}}(D_s(z))\|^2, \tag{45}$$

$$\Phi_{\text{obj}}(z) = \sum_i \sigma_{t,i} J_i(D_s(z), D_U(z), p_t), \tag{46}$$

$$V_t(z) = \epsilon_{\text{snap}}^{-1}\Phi_{\text{meas}}(z) + \Phi_{\text{obj}}(z), \qquad \mathcal{F}_t(z) = -M(z)^{-1}\nabla_z^\circ V_t(z), \quad M(z) \succ 0. \tag{47}$$

Here ∇_z° is the ordinary gradient at smooth points and the selected automatic-differentiation subgradient at nonsmooth kinks. The identity choice $M = I$ recovers the unpreconditioned formulation; the implemented RK2 power-flow field uses $M = G$, including its configured step gain before velocity capping. The current measurement and filtered priority are held fixed while taking the within-cycle derivative. Differentiating the state-dependent priority again would define a different vector field. The frozen decoders make this an online latent response update rather than a policy-training step.

The terminology "snap" and "slide" describes intended operation: when reconstruction error is large, the heavily weighted restoration component dominates; after consistency improves, the objective component shapes the response. A small $\epsilon_{\text{snap}} > 0$ increases stiffness and does not permit an arbitrarily large explicit integration step. Exact normal/tangential decomposition, contraction rates, and second-order integration claims require the additional assumptions established in the theory appendix. The energy design alone is not a global optimization proof.

E.3 Capped Midpoint RK2

For latent step size h_z and cap $V_{\max}$, define

$$\overline{\mathcal{F}}_t(z) = \frac{\mathcal{F}_t(z)}{\max(1, \|\mathcal{F}_t(z)\|/V_{\max})}. \tag{48}$$

The midpoint predictor evaluates

$$k_1 = \overline{\mathcal{F}}_t(z_t^0), \quad z_{1/2} = z_t^0 + \tfrac{1}{2}h_z k_1, \quad k_2 = \overline{\mathcal{F}}_t(z_{1/2}), \quad d_E = h_z k_2. \tag{49}$$

The physical interval Δt advances the plant; h_z advances the latent response. They can be coupled by a controller configuration but are not interchangeable in the numerical analysis. There are two field-gradient evaluations, not one. The cap bounds $\|d_E\| \le h_z V_{\max}$; it does not by itself bound the complete step after geometric correction and retraction.

For smooth uncapped fields, the midpoint method has its standard second-order finite-interval accuracy. A radial cap is continuous but may be nonsmooth at its activation boundary. A uniform second-order proof over a trajectory crossing that boundary requires additional regularity or a separate piecewise analysis; the deployment tube theorem instead uses the actual accepted displacement.

E.4 Decoder Metric and Lie-Local Residual

The local geometry is described by

$$G(z) = J_{D_s}(z)^\top J_{D_s}(z) + \lambda_m I, \tag{50}$$

which is positive definite for $\lambda_m > 0$. Retain the top-k eigenvectors B_G and collect their positive eigenvalues in $\boldsymbol{\lambda}$. The local step $d = B_G^\top d_E$ and scaled direction $v = (B_G^\top k_2) \oslash \sqrt{\boldsymbol{\lambda}}$ determine three-dimensional rotations. For a block $v_b = (v_1, v_2, v_3)$, define

$$[v_b]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \qquad R_b = \exp(s_L h_z [v_b]_\times). \tag{51}$$

The residual is

$$d_L = B_G\left[\text{blockdiag}(R_b)d - d\right]. \tag{52}$$

A two-dimensional final block is padded with a zero third coordinate in both the step and generator, rotated in three dimensions, and truncated to its first two coordinates. A one-dimensional final block is unchanged. The full three-dimensional rotation preserves norm; padding and truncation alone do

not make the induced two-dimensional map a norm-preserving rotation. The residual is restricted to the retained decoder-sensitive directions; a non-finite residual is replaced by zero. The correction strength γ_L scales it relative to the RK2 predictor. The same decoder metric can also precondition the field as $-G^{-1}\nabla V_t$, as in the implemented power-flow RK2 controller. The Euclidean-field variant uses the identity preconditioner.

Each block rotation preserves its vector norm. This is an algebraic property of the local rotation, not a certificate for decoded physical feasibility or invariance of $\mathcal{M}^*$. The relevant Lie-exponential and displacement bounds, including their constants and domain assumptions, are given in the theory appendix.

E.5 Retraction, Final Output, and Computational Cost

With group/chart composition $\oplus$, compute

$$\bar{z} = z_t^0 \oplus (d_E + \gamma_L d_L), \qquad z_t^+ = (1-\mu_R)\bar{z} + \mu_R \widehat{\mathrm{proj}}_{\mathcal{M}^*}(\bar{z}). \tag{53}$$

In a Euclidean chart, $\oplus$ is addition. The implemented $\widehat{\mathrm{proj}}_{\mathcal{M}^*}$ is the indexed nearest-reference query defined in the main-text remark, rather than an exact projection onto an implicit nonlinear manifold. Setting $\gamma_L = \mu_R = 0$ gives plain RK2. A mixed reference correction can reduce drift empirically, but it is not automatically nonexpansive and need not lie on the underlying continuous set. The theory's accepted-step bound must cover the actual map.

The response decoder returns $U_t = D_U(z_t^+)$. For box constraints, clamp its control coordinates; for a plan, enforce any configured inter-step limits in temporal order using the actual previous action. Evaluate all other physical constraints on this final response. Execute the first action and use the next measurement to start a new cycle. Clipping enforces its own bounds but can change coupled constraints, so pre-clipping feasibility is not substituted for final-response feasibility.

The explicit update uses two field gradients, a decoder Jacobian, and a top-eigenspace calculation, followed by block exponentials and any projection. Constructing the full metric costs at least the matrix operations associated with the decoder Jacobian, and a dense eigendecomposition is on the latent dimension. It is not generally a $k \times k$ eigensolve merely because only k directions are retained. Reduced eigenspace implementations or metric caching alter this cost and must be timed as their own configurations.

F Action Decoder Training Pipeline

F.1 Why Pointwise Supervision Is Insufficient

Pointwise supervision trains at $E_x(x_k)$, while deployment can decode at an intermediate point created by integration, correction, and retraction. These two latent distributions need not agree. Stage II therefore adds synthetic intermediate queries and, when available, evaluates them with the multi-step physical terms in (6).

In the transition-indexed variant, the encoders, state decoder, and geometric embedding are held fixed; only D_U is optimized. A stored transition comprises $(z_h, z_{h+1}, U_h^{\mathrm{ref}}, U_{h+1}^{\mathrm{ref}}, p_h, p_{h+1})$. Define $\Delta U_h^{\mathrm{ref}} = U_{h+1}^{\mathrm{ref}} - U_h^{\mathrm{ref}}$ and $\Delta p_h = p_{h+1} - p_h$. The risk/transition weight is

$$\omega_h = \frac{\alpha_u \|\Delta U_h^{\mathrm{ref}}\| + \alpha_p \|\Delta p_h\|}{\mathbb{E}_{\mathcal{B}}[\alpha_u \|\Delta U^{\mathrm{ref}}\| + \alpha_p \|\Delta p\|] + \varepsilon_\omega}. \tag{54}$$

The positive stabilizer handles minibatches whose transitions are constant. This weighting emphasizes dynamically demanding transitions. Larger action or parameter increments are proxies for difficulty, not physical certificates.

F.2 Endpoint, Increment, and Intermediate Targets

For response-target and flow interpolation fractions $\alpha_r, \alpha_f \in (0, 1]$, let $z_h^{\alpha_r} = z_h + \alpha_r(z_{h+1} - z_h)$ and $z_h^{\alpha_f} = z_h + \alpha_f(z_{h+1} - z_h)$. The endpoint loss is

$$\ell_{\mathrm{end}} = \|D_U(z_h) - U_h^{\mathrm{ref}}\|^2 + \|D_U(z_{h+1}) - U_{h+1}^{\mathrm{ref}}\|^2. \tag{55}$$

The increment loss preserves the reference change in action:

$$\ell_\Delta = \|D_U(z_{h+1}) - D_U(z_h) - \Delta U_h^{\mathrm{ref}}\|^2. \tag{56}$$

Two intermediate-target constructions are distinguished. A next-target term uses

$$\ell_{\mathrm{next}} = \|D_U(z_h^{\alpha_r}) - U_{h+1}^{\mathrm{ref}}\|^2, \tag{57}$$

whereas an interpolation/flow term uses

$$\ell_{\mathrm{flow}} = \|D_U(z_h^{\alpha_f}) - (U_h^{\mathrm{ref}} + \alpha_f \Delta U_h^{\mathrm{ref}})\|^2. \tag{58}$$

The first encourages the intermediate code to produce the next desired response; the second encourages smooth progress between endpoint responses. They are different supervision choices and should not be conflated. In plan-valued domains, Π_h extracts the corresponding horizon action whenever the tuple target is an action rather than the whole plan. Interpolated targets are not asserted to satisfy nonconvex physical constraints.

For physical rollout supervision, every query-target pair must also carry an aligned reference initial state and parameter sequence. The two endpoint tuples use $(z_h, U_h^{\mathrm{ref}}, s_h, p_{h:h+K})$ and $(z_{h+1}, U_{h+1}^{\mathrm{ref}}, s_{h+1}, p_{h+1:h+K+1})$. The next-target and Lie-corrected tuples use $(z_h^{\alpha_r}, U_{h+1}^{\mathrm{ref}}, s_{h+1}, p_{h+1:h+K+1})$ and $(\tilde{z}_{h+1}, U_{h+1}^{\mathrm{ref}}, s_{h+1}, p_{h+1:h+K+1})$, respectively. A flow-interpolated query receives physical rollout terms only when an aligned reference rollout is generated offline; otherwise it receives response-level supervision only. The increment loss above and any inter-step control-smoothness penalty are optional regularizers, not part of the core query–rollout coupling in (6).

F.3 Lie-Corrected Surrogate Latents

Construct a transition step $d_E = z_h^{\alpha_r} - z_h$. At z_h, use the decoder metric and generator $s_L[(B_G^\top d_E) \oslash \sqrt{\boldsymbol{\lambda}}]_\times$ to obtain d_L. No additional h_z multiplies this training generator: d_E is already a displacement. Online the generator uses $s_L h_z[(B_G^\top k_2) \oslash \sqrt{\boldsymbol{\lambda}}]_\times$, so the two conventions agree when $d_E = h_z k_2$. The surrogate latent is

$$\tilde{z}_{h+1} = z_h \oplus (d_E + \gamma_L d_L). \tag{59}$$

The online projection mixture is not applied to this synthetic training point. At deployment, the corresponding local Lie step is a fast, solver-free feasibility correction inside the displacement guard; Proposition 12 states the conditions under which the accepted update preserves physical feasibility. The same local-frame scales and residual strength should be used in training. Supervision at this point includes

$$\ell_{L,U} = \|D_U(\tilde{z}_{h+1}) - U_{h+1}^{\mathrm{ref}}\|^2, \tag{60}$$

$$\ell_{L,\Delta} = \|D_U(\tilde{z}_{h+1}) - D_U(z_h) - \Delta U_h^{\mathrm{ref}}\|^2. \tag{61}$$

Equation (7) uses $\Phi_K(\tilde{z}_{h+1}, U_{h+1}^{\mathrm{ref}}; \mathcal{R}_{h+1})$ as its core Lie branch. An implementation may additionally use $\lambda_\Delta \ell_{L,\Delta}$; setting $\lambda_L = 0$ omits the branch, while setting $\lambda_R = 0$ recovers response-level transition refinement.

F.4 Frozen Geometry and Latent Diagnostics

The pipeline also tracks latent anchoring and residual-norm terms,

$$\ell_{L,\mathrm{anchor}} = \|\tilde{z}_{h+1} - \mathrm{stopgrad}(z_h^{\alpha_r})\|^2, \qquad \ell_{L,\mathrm{norm}} = \|d_L\|^2. \tag{62}$$

When the encoders, state decoder, and geometry are all frozen, neither term depends on D_U. Their gradients with respect to decoder parameters are zero. They remain useful diagnostics of surrogate quality; they become optimization regularizers only in a separately specified stage that trains the geometric modules.

The latent-query terms are closed-loop-aware but not closed-loop-equivalent: they use oracle transitions and synthetic intermediate latents. The multi-step terms roll one decoded horizon plan through the known dynamics, but do not simulate repeated closed-loop re-encoding and replanning. The theoretical appendix states the additional coupling conditions needed to turn latent-distribution proximity into a deployment supervision bound. A small synthetic training loss alone is not such a coupling certificate.

F.5 Navigation's Learned Correction and Candidate Refinement

The evaluated navigation pipeline instantiates the same response-map idea with a different online update. Its observation encoder receives a normalized scene representation, optionally with graph features, and a plan head returns the complete joint horizon plan. During rollout-aware refinement, this plan is propagated for $K = H$ steps through differentiable joint dynamics. The loss compares the predicted plan, joint state trajectory, and dynamic interaction summary with their references, includes the margin terms in (6), and regularizes inter-step control differences. The observation encoder, plan head, and safety head are refined in this stage, while the reference and state decoders remain frozen. A learned correction network then consumes the latent code and nominal plan and predicts an additive plan correction. The navigation controller evaluates these networks without computing the explicit RK2 field online.

The corrected plan is the anchor for refinement. A sequential projection clamps each horizon action against the physical box and the previous projected action, starting from the actual u_{t-1}. Candidate initialization includes the projected learned anchor, a goal-directed reactive plan, an aggressive separation plan, safety templates, per-robot maneuvers, and mixtures of the anchor with reactive/separation plans (Khatib, 1986; Koditschek, 1987; Chang & Howes, 2012). Structured noise perturbs active robot controls around these bases and the current best plan; every candidate is projected before rollout scoring. The evaluated candidate counts are reported with the navigation experiments; the GPC and reference refinements do not use equal end-to-end budgets.

For predicted steps $k = 0, \dots, H-1$ with margins $m_{k,j}$, the controller computes a weighted safety score containing violations, near-boundary penalties, and a discrete margin-change residual:

$$\mathcal{B}_{\mathrm{safe}}(U) = \sum_{k=0}^{H-1} a_k \Big[\sum_j [-m_{k,j}]_+^2 + 0.25 \sum_j [0.20 - m_{k,j}]_+^2 + 0.5\, \mathbf{1}_{k\geq 1} \sum_j [0.65 m_{k-1,j} - m_{k,j}]_+^2 \Big], \qquad a_k = 1 + 0.15(H-k), \quad (63)$$

where $[x]_+ = \max(x, 0)$. Additional terms penalize the minimum horizon margin, and the scored cost adds task cost and a trust penalty proportional to mean squared distance from the learned anchor. The implemented comparator is lexicographic: a predicted-feasible candidate outranks an infeasible one; among feasible candidates, lower scored cost wins; among infeasible candidates, lower safety penalty, then higher minimum margin, then lower cost win. A finite search can still return an infeasible response when no feasible candidate is found; its empirical success does not become an unconditional guarantee.

The first action of the selected joint plan is executed, and the observation is rebuilt at the next cycle. The navigation evidence concerns this joint-plan pipeline: the learned map, learned correction, and explicit finite refinement together. The RK2/Lie update in Section 3.5 is a separate latent-flow instantiation for tightly constrained optimization and is not inferred from navigation success.

G From Decoder Training to Deployment Error

The latent-query terms in the second training stage change where the response decoder is supervised. Static reconstruction uses encoded dataset points, whereas deployment evaluates the decoder after localization, geometric correction, or candidate refinement. This section gives their distribution-transfer argument; the multi-step physical terms in (6) and the per-cycle feasibility theorem address different questions. The results apply to the response map actually being evaluated; for a single executed action, replace the plan decoder by its first-action projection. A numerical reference is not assumed to be globally optimal.

G.1 Distributions and the comparator

Fix a context and let $\mathcal{Z}$ be a compact latent domain with Euclidean distance. Let $D : \mathcal{Z} \to \mathbb{R}^m$ denote the frozen decoder and $U^{\mathrm{ref}} : \mathcal{Z} \to \mathbb{R}^m$ a specified, single-valued reference selection. Define

$$e(z) = \|D(z) - U^{\mathrm{ref}}(z)\|. \quad (64)$$

The three distributions of interest are the encoded training marginal ρ_{off}, the synthetic rollout marginal ρ_{syn}, and the actual post-update deployment marginal ρ_{on}. They need not coincide. In particular, oracle-transition interpolation does not generate the physical-state distribution induced by recursively executing the learned controller.

For distributions ρ, ν on $\mathcal{Z}$, let $\Pi(\rho, \nu)$ be the set of their joint couplings and define

$$W_1(\rho, \nu) = \inf_{\pi \in \Pi(\rho,\nu)} \int_{\mathcal{Z}\times\mathcal{Z}} \|z - z'\| \, d\pi(z, z'). \tag{65}$$

Compactness ensures finite integrals. The same arguments hold on a noncompact domain when the relevant first moments and expected errors are finite.

Proposition 16 (Decoder error under a change of latent distribution)**.** *Suppose D and the selected reference map are Lipschitz on $\mathcal{Z}$, with constants K_D and K_{ref}. Then e is Lipschitz with constant $L_e \le K_D + K_{\text{ref}}$, and*

$$|\mathbb{E}_\rho e - \mathbb{E}_\nu e| \le L_e W_1(\rho, \nu). \tag{66}$$

Consequently,

$$\mathbb{E}_{\rho_{\text{on}}} e \le \mathbb{E}_{\rho_{\text{off}}} e + L_e W_1(\rho_{\text{on}}, \rho_{\text{off}}). \tag{67}$$

Proof. The reverse triangle inequality followed by the two Lipschitz bounds gives

$$\begin{aligned} |e(z) - e(z')| &\le \|(D(z) - U^{\text{ref}}(z)) - (D(z') - U^{\text{ref}}(z'))\| \\ &\le \|D(z) - D(z')\| + \|U^{\text{ref}}(z) - U^{\text{ref}}(z')\| \\ &\le (K_D + K_{\text{ref}})\|z - z'\|. \end{aligned} \tag{68}$$

For any $\pi \in \Pi(\rho, \nu)$, its two marginals imply

$$\begin{aligned} |\mathbb{E}_\rho e - \mathbb{E}_\nu e| &= \left| \int (e(z) - e(z')) \, d\pi(z, z') \right| \\ &\le L_e \int \|z - z'\| \, d\pi(z, z'). \end{aligned}$$

Taking the infimum over all couplings proves (66); setting $\rho = \rho_{\text{on}}$ and $\nu = \rho_{\text{off}}$ gives (67). □

The reference regularity is substantive: a discontinuous switch between nonconvex solution branches can make K_{ref} unavailable even when the neural decoder is smooth. For a set-valued reference family, a particular selection must be specified before using this proposition. The next result avoids a Lipschitz reference-map assumption by keeping the target paired with each training tuple.

G.2 Paired-target transfer for synthetic rollout supervision

Let $\widetilde{Z}$ be a synthetic rollout code, Z^+ its coupled post-update deployment code, and Y the response target assigned to that same transition. The coupling must pair codes and targets under the same context and transition convention. Comparing unrelated latent marginals while ignoring target assignments is insufficient.

Proposition 17 (Transfer with a shared response target)**.** *Suppose D is K_D-Lipschitz and the joint distribution of $(Z^+, \widetilde{Z}, Y)$ satisfies $\mathbb{E}\|Z^+ - \widetilde{Z}\| \le \varepsilon_{\text{syn}}$. Then*

$$\left| \mathbb{E}\|D(Z^+) - Y\| - \mathbb{E}\|D(\widetilde{Z}) - Y\| \right| \le K_D \varepsilon_{\text{syn}}. \tag{69}$$

In particular, if $\mathcal{L}_{\text{syn}} = \mathbb{E}\|D(\widetilde{Z}) - Y\|^2$, then

$$\mathbb{E}\|D(Z^+) - Y\| \le \sqrt{\mathcal{L}_{\text{syn}}} + K_D \varepsilon_{\text{syn}}. \tag{70}$$

If deployment instead has a paired target Y^+ with $\mathbb{E}\|Y^+ - Y\| \le \delta_Y$, the right side increases by δ_Y.

Proof. For each sample of the joint distribution,

$$\big|\|D(Z^+) - Y\| - \|D(\widetilde{Z}) - Y\|\big| \le \|D(Z^+) - D(\widetilde{Z})\| \le K_D\|Z^+ - \widetilde{Z}\|.$$

Taking expectations proves (69). Cauchy–Schwarz gives $\mathbb{E}\|D(\widetilde{Z}) - Y\| \le \sqrt{\mathcal{L}_{\mathrm{syn}}}$, which proves (70). For a changed target, use $\|D(Z^+) - Y^+\| \le \|D(Z^+) - Y\| + \|Y - Y^+\|$ and take expectations. □

The coefficient in (69) is K_D, not $K_D + K_{\mathrm{ref}}$, because the two errors use the same target. Conversely, this result does not show closeness to an independently reoptimized current-priority response unless that response is the paired Y, or its mismatch is bounded by δ_Y. Constructing synthetic Lie-corrected codes supplies a candidate coupling; it does not by itself establish a small $\varepsilon_{\mathrm{syn}}$ for the deployed controller.

G.3 Risk weighting and the quantity controlled by training

The decoder objective upweights transitions with large response or disturbance changes. This changes the measure under which training error is averaged. Let P be a distribution of paired synthetic transitions and let $\omega \ge 0$ satisfy $\mathbb{E}_P\omega = 1$. The weighted probability distribution is $dP_\omega = \omega\, dP$, and

$$\mathcal{L}_\omega = \mathbb{E}_P\big[\omega\|D(\widetilde{Z}) - Y\|^2\big] = \mathbb{E}_{P_\omega}\|D(\widetilde{Z}) - Y\|^2. \tag{71}$$

Thus $\sqrt{\mathcal{L}_\omega}$ bounds the mean error under P_ω. It cannot automatically replace $\sqrt{\mathcal{L}_{\mathrm{syn}}}$ under P. If $\omega \ge \omega_{\min} > 0$ almost surely, the corresponding unweighted bound is

$$\mathbb{E}_P\|D(\widetilde{Z}) - Y\| \le \sqrt{\mathbb{E}_P\|D(\widetilde{Z}) - Y\|^2} \le \sqrt{\mathcal{L}_\omega/\omega_{\min}}. \tag{72}$$

The first inequality is Cauchy–Schwarz; the second follows pointwise from $\omega_{\min}\|D(\widetilde{Z}) - Y\|^2 \le \omega\|D(\widetilde{Z}) - Y\|^2$. Without a positive lower bound, transitions assigned zero or arbitrarily small weight need separate error control. The minibatch-normalized weighting rule is an empirical objective; transferring its observed loss to a population loss additionally needs a generalization argument. None of the inequalities above substitutes a training-set average for a population guarantee without that step.

G.4 Uniform bounds across feedback cycles

The measured context and the reference selection change with time. A uniform claim requires assumptions that hold across those contexts, rather than a proof at one fixed operating condition.

Proposition 18 (Uniform training-to-deployment transfer)**.** *For each cycle t, let D_t be a K-Lipschitz response map on the relevant paired supports. Suppose*

$$\sup_t \mathbb{E}\|Z_t^+ - \widetilde{Z}_t\| \le \varepsilon_{\mathrm{syn}}, \qquad \sup_t \mathbb{E}\|D_t(\widetilde{Z}_t) - Y_t\|^2 \le \overline{\mathcal{L}}. \tag{73}$$

Then

$$\sup_t \mathbb{E}\|D_t(Z_t^+) - Y_t\| \le \sqrt{\overline{\mathcal{L}}} + K\varepsilon_{\mathrm{syn}}. \tag{74}$$

Alternatively, if e_t is uniformly L_e-Lipschitz and $\sup_t W_1(\rho_{\mathrm{on},t}, \rho_{\mathrm{syn},t}) \le \varepsilon_{\mathrm{syn}}$, then

$$\sup_t \big|\mathbb{E}_{\rho_{\mathrm{on},t}} e_t - \mathbb{E}_{\rho_{\mathrm{syn},t}} e_t\big| \le L_e\varepsilon_{\mathrm{syn}}. \tag{75}$$

Proof. At each fixed t, apply Proposition 17 to the joint distribution in (73). Its right side is bounded by the same two constants at every cycle. Taking the supremum gives (74). For the second assertion, apply Proposition 16 at each t and again take the supremum. No induction over previous latent iterates is used. □

Observation reset motivates constructing and testing these couplings separately at each cycle. It does not prove their uniform bounds. In particular, fresh localization prevents a previous numerical latent error from being mechanically reused, but physical-state errors can still affect later observations, constraints, and target responses. The results above concern response-supervision error. Feasibility needs the margins and final-output conditions of Theorem 1; physical closed-loop stability or accumulated task regret needs additional dynamical assumptions.

H EVALUATION PROTOCOL

Table 4: Empirical scope and principal physical coupling.

Setting	Training / range	Evaluation	Coupled constraints
Analytical	Two objective landscapes	100 trajectories per variant	Dynamics and input slew
Static navigation	60,000 references	100 episodes, 150 steps	Joint scene safety
Moving navigation	120,000 references	100 episodes, 600 steps	Obstacle motion and joint safety
Nominal OPF	5000 loads $\times$ 50 weights	300 dispatch steps; GPC on 100 load sequences (Table 7)	AC balance and ramps
Perturbed OPF	Fixed branch realization per trajectory	300 dispatch steps, 299 finite costs; GPC on 100 load sequences (Table 7)	Same limits with parameter variation
$N-k$ OPF	1,001 train graphs, $N-0$ through $N-5$	GPC on one held-out $N-5$ graph, 60 steps	AC balance, ramps, and topology shift

For perturbed OPF, objective means use 299 finite costs and overall constraint feasibility, including the generator diagnostic, is $299/300$. The $N-k$ extension evaluates GPC against a paired numerical reference; its timing excludes the first CUDA/model-initialization step.

The nominal cumulative GPC cost and MO-IPOPT gap are 108324.4387 and 325.1895, yielding a reference-relative gap of approximately 0.30%. The perturbed cumulative values over finite costs are 88293.2920 and 220.0229, yielding approximately 0.25%. The comparator is a numerical nonconvex reference, not a certified global optimizer.

Analytical means, standard errors, safe fractions, slew fractions, and failures come from 100-trajectory batch summaries. Obstacle and slew violation percentages are 100 times one minus the corresponding mean safe fraction. Trajectory failure is a separate indicator. Navigation uses the reported 100-episode evaluation; its moving GPC success rate is reachable-conditioned after excluding a reference-unreachable scene. It is therefore kept distinct from unconditional reference success. Representative trajectories illustrate mechanisms, not independent aggregate estimates.

The reported RL budget is 10^6 environment interactions. Navigation compares PPO, PPO-Lagrangian, CPO, and one-step shielded PPO. OPF compares TD3, CPO, and linearized receding-horizon MPC. GPC receives uncertainty-aware training and the branch descriptor in perturbed OPF, whereas TD3/CPO retain nominal training. Navigation uses 4096 GPC refinement candidates and 384 reference candidates. These comparisons evaluate the specified deployed controllers and training protocols; the perturbed OPF result tests GPC on an unseen realization within its modeled parameter family without retraining.

I MULTI-AGENT NAVIGATION

I.1 DYNAMICS AND HARD CONSTRAINTS

The ego robot is indexed by 0 and up to $N \leq 5$ neighbors by $1, \ldots, N$, with active masks m_i and $m_0 = 1$. Controls are acceleration and turn rate, $u^i = (a^i, \omega^i)$. The detailed midpoint unicycle model is

$$\theta^i_{t+k+1} = \theta^i_{t+k} + \Delta t\, \omega^i_{t+k}, \tag{76}$$

$$v^i_{t+k+1} = \mathrm{clip}(v^i_{t+k} + \Delta t\, a^i_{t+k}, v_{\min}, v_{\max}), \tag{77}$$

$$p^i_{t+k+1} = p^i_{t+k} + \Delta t\, \bar{v}^i_{t+k}(\cos\bar{\theta}^i_{t+k}, \sin\bar{\theta}^i_{t+k}), \tag{78}$$

where $\bar{v}^i_{t+k} = (v^i_{t+k} + v^i_{t+k+1})/2$ and $\bar{\theta}^i_{t+k} = \theta^i_{t+k} + \Delta t\, \omega^i_{t+k}/2$. Controls obey acceleration, angular-velocity, and slew limits. Obstacle centers advance by current velocity with boundary re-

flection. All active horizon margins satisfy

$$h^{\mathrm{obs}}_{i,j}(t+k) = \|p^i_{t+k} - c^j_{t+k}\| - r_i - r_j - d_{\mathrm{safe}} \geq 0, \tag{79}$$

$$h^{\mathrm{rr}}_{i,\ell}(t+k) = \|p^i_{t+k} - p^\ell_{t+k}\| - r_i - r_\ell - d_{\mathrm{safe}} \geq 0. \tag{80}$$

The reported settings are $H = 8$, $\Delta t = 0.2$, $d_{\mathrm{safe}} = 0.18$, up to 15 obstacles, and ego/neighbor goal weighting $\eta_g = 0.65$.

Control bounds and horizon consistency. The acceleration, angular-rate, and slew constraints are

$$a_{\min} \leq a^i_{t+k} \leq a_{\max}, \qquad |\omega^i_{t+k}| \leq \omega_{\max}, \qquad \|u^i_{t+k} - u^i_{t+k-1}\|_\infty \leq \Delta u_{\max}. \tag{81}$$

The first slew difference uses the actual previously executed action. Future differences use adjacent actions in the candidate plan. Moving obstacles follow $c^j_{t+k+1} = c^j_{t+k} + \Delta t\, v^j_{o,t+k}$ with boundary reflection. Static obstacles are the zero-velocity case. The same modeled horizon and physical margins are used for reference generation and candidate evaluation, while their search procedures and computational budgets differ.

At each cycle, the learned response and any refinements represent all active agents' controls over the horizon. Only the first joint action is applied; future actions are replaced after observing the updated scene. This separates the finite-horizon prediction problem from the longer closed-loop evaluation, which lasts 150 steps in the static benchmark and 600 steps in the moving benchmark.

I.2 Six-Term Objective

Write $n_a = 1 + \sum_{i=1}^N m_i$ and

$$G_k = \|p^0_{t+k} - g^0\|^2 + \eta_g \frac{\sum_{i=1}^N m_i \|p^i_{t+k} - g^i\|^2}{\max(\sum_{i=1}^N m_i, 1)}. \tag{82}$$

For the set $\mathcal{H}_{t+k}$ of active margins, the six terms are

$$J_{\mathrm{goal}} = \sum_{k=1}^H G_k, \qquad J_{\mathrm{term}} = G_H, \tag{83}$$

$$J_{\mathrm{safe}} = \sum_{k=1}^H \mathrm{mean}_{h\in\mathcal{H}_{t+k}}\, e^{-\mathrm{clip}(h,-3,3)}, \qquad J_{\mathrm{viol}} = \sum_{k=1}^H \sum_{h\in\mathcal{H}_{t+k}} [\max(0,-h)]^2, \tag{84}$$

$$J_{\mathrm{effort}} = \sum_{k=0}^{H-1} \frac{\sum_{i=0}^N m_i \|u^i_{t+k}\|^2}{n_a}, \qquad J_{\mathrm{smooth}} = \frac{1}{H} \sum_{k=0}^{H-1} \sum_{i=0}^N m_i \|u^i_{t+k} - u^i_{t+k-1}\|^2. \tag{85}$$

The cost is $w_1\lambda_g J_{\mathrm{goal}} + w_2\lambda_s J_{\mathrm{safe}} + w_3\lambda_e J_{\mathrm{effort}} + \lambda_v J_{\mathrm{viol}} + \lambda_T J_{\mathrm{term}} + \lambda_\Delta J_{\mathrm{smooth}}$, with coefficient tuple $(2, 12, 0.08, 260, 5, 0.10)$. Only the first three terms vary with $w \in \Delta^2$. The violation penalty does not replace hard-constraint evaluation.

I.3 Closed-Loop Metrics and Reference Conditioning

For episode e, define the minimum observed margins

$$m^{(e)}_{\mathrm{obs}} = \min_{t,i,j} h^{\mathrm{obs}}_{i,j}(t), \qquad m^{(e)}_{\mathrm{pair}} = \min_{t,i<\ell} h^{\mathrm{rr}}_{i,\ell}(t). \tag{86}$$

Scene collision freedom requires both to be nonnegative. Joint success requires all active agents to reach their goals within the configured goal radius. Safe success is the conjunction. A scene can therefore be collision-free but fail by not finishing, or finish while failing the safety criterion.

The reference-safe set is

$$\mathcal{I}_{\mathrm{ref}} = \{e : \mathrm{joint}^{(e)}_{\mathrm{ref}} = 1, m^{(e)}_{\mathrm{obs,ref}} \geq 0,\ m^{(e)}_{\mathrm{pair,ref}} \geq 0\}. \tag{87}$$

Cost gaps are aggregated over this set, rather than interpreting an unsafe low-cost reference trajectory as an optimum. In moving scenes the reference (the multistart NMPC in Table 1) has 83% joint

success, 42% collision freedom, and 35% safe success. After excluding one designated reference-unreachable scene from the 100-scene set, the reachability-conditioned safe-success rate over the remaining 99 scenes is reported as 99% in Table 1. Scene collision freedom holds throughout the included evaluation.

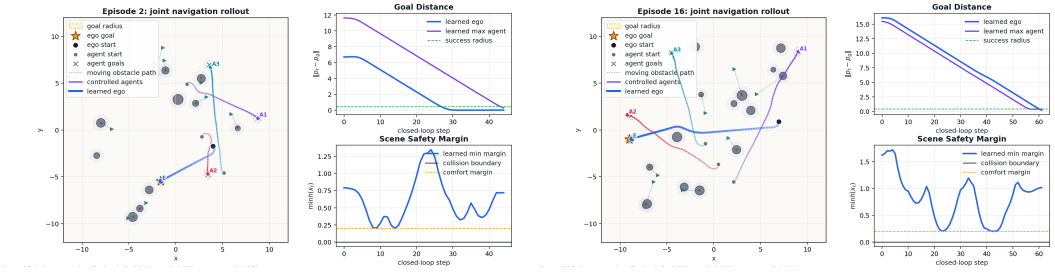
The nonzero positive margin summaries provide additional geometric information but should not be substituted for explicit episode counts. Their aggregation over per-episode minima is different from a uniform lower bound on unseen scenes. Likewise, a negative reference-conditioned cost gap means improvement over that finite-budget reference on its successful subset, not performance better than a global optimum.

I.4 Additional Navigation Results

In a paired moving-scene control with the same nominal candidate count (384) for GPC and multistart NMPC, GPC attains 96/100 safe successes versus 35/100 for NMPC. Among the 63 scenes with different outcomes, GPC alone succeeds in 62 and NMPC alone in one. This control uses the full 100-scene set and is separate from the reachable-conditioned main-table rate.

Table 5: Reported aggregate per-episode minimum-margin summaries.

Setting	Scene	Ego	Agent–obstacle	Agent–agent
Static	0.301	0.900	0.503	2.477
Moving	0.203	0.528	0.253	1.065



(a) Moving episode 2

(b) Moving episode 16

Figure 4: Additional held-out moving-obstacle rollouts showing the joint scene, goal-distance traces, and minimum safety-margin histories.

I.5 Recorded Moving-Scene Non-Completion Failure

Figure 5 documents the recorded moving-scene failure in episode 24. The rollout remains collision-free, with minimum scene margin 0.200, and the ego reaches its goal with terminal distance 0.012. Joint completion nevertheless fails because the farthest controlled agent remains 5.887 from its goal over the plotted first 100 steps; joint completion is judged over the full 600-step episode. The goal-distance plateau therefore identifies incomplete agent arrival, rather than collision, as the failure mode.

J Optimal Power Flow

The main IEEE 30-bus AC network contains coupled nonlinear balance equations, generator capacity, voltage and thermal limits, and temporal dispatch ramps. The physical state is the bus voltage-phasor vector (V, θ); the controllable dispatch contains active and reactive generation, with voltage-control variables supplied by the chosen solver/controller interface. Load is exogenous. The measured-state input is augmented by the available load and parameter descriptors. In the parameter-varied experiment, the branch realization is supplied to the controller; it is not an unobserved disturbance that the experiment asks GPC to identify. Section J.5 then changes both scale and graph structure, using an IEEE 118-bus family with line-outage contingencies.

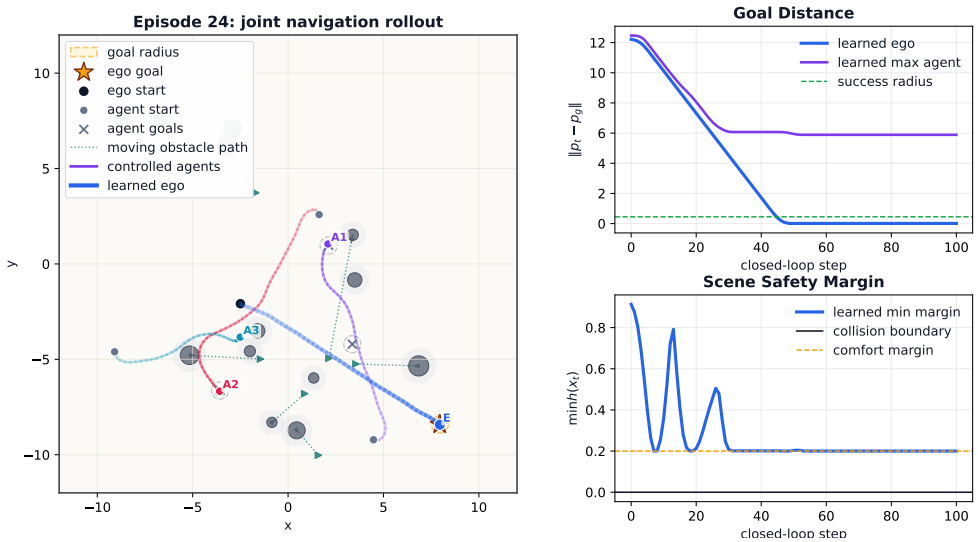


Figure 5: Recorded moving-scene GPC failure (episode 24; first 100 of 600 closed-loop steps). Left: joint trajectories. Upper right: ego and worst-agent goal distances. Lower right: minimum scene safety margin. The ego reaches its goal and the safety margin never crosses the collision boundary, but another controlled agent remains outside the goal region.

J.1 Network Equations and Branch Ratings

At bus i, with voltage V_i, angle θ_i, and network admittance $G + \mathrm{j}B$, AC balance requires

$$P_{G,i} - P_{D,i} = V_i \sum_j V_j [G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)], \tag{88}$$

$$Q_{G,i} - Q_{D,i} = V_i \sum_j V_j [G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j)]. \tag{89}$$

Other constraints are generator capacity, voltage bounds, endpoint thermal ratings, and ramps $|P_{G,i,t} - P_{G,i,t-1}| \le R_i$. For a branch $\ell = (i, j)$, let $\mathcal{V}_i = V_i e^{\mathrm{j}\theta_i}$. With its branch admittance blocks, the complex endpoint powers are

$$S_{ij} = \mathcal{V}_i \overline{Y_{ff}\mathcal{V}_i + Y_{ft}\mathcal{V}_j}, \qquad S_{ji} = \mathcal{V}_j \overline{Y_{tf}\mathcal{V}_i + Y_{tt}\mathcal{V}_j}. \tag{90}$$

Define $\bar{S}_\ell = \max\{|S_{ij}|, |S_{ji}|\}$ and impose $\bar{S}_\ell \le S_\ell^{\max}$. This form accommodates transformer taps, phase shifts, and shunts represented in the network model. Power-flow residuals are evaluated separately from inequality margins; a converged state calculation does not imply that generator limits also hold.

J.2 Objective Hierarchy and Semantic Indicators

The objectives are

$$J_{\text{thermal}} = \sum_\ell [\max(0, \bar{S}_\ell / S_\ell^{\max} - 0.85)]^4, \tag{91}$$

$$J_{\text{voltage}} = \sum_i [\max(0, |V_i - 1| - 0.05)]^2, \tag{92}$$

$$J_{\text{econ}} = \sum_i (a_i P_{G,i}^2 + b_i P_{G,i} + c_i). \tag{93}$$

The urgency indicators use normalized maximum branch loading and worst voltage deviation:

$$\delta_{\text{th}} = \text{clip}\left(\max_\ell \frac{\bar{S}_\ell}{S_\ell^{\max}}, 0, 1\right), \qquad \delta_v = \text{clip}\left(\frac{\max_i |V_i - V_{\text{nom}}|}{\Delta V_{\text{limit}}}, 0, 1\right). \tag{94}$$

The detailed formulation uses a small positive constant economic indicator. Nominal economic dominance depends on the relative scaling of these indicators and their exponential urgency parameters.

The thermal quartic activates near a loading ratio of 0.85, before the hard rating is reached. The voltage objective uses a 0.05 p.u. deadband around nominal voltage. Economic cost uses generator-specific quadratic coefficients. These channels express different operating priorities: urgent physical recovery and slower economic adjustment. Their numerical scales must be calibrated before the exponential urgency map is interpreted; changing units without changing indicator normalization would change the controller.

For a ramp-coupled horizon, the scalarized problem can be written

$$\min_{\{P_{G,h},Q_{G,h},V_h,\theta_h\}_{h=0}^{H-1}} \sum_{h=0}^{H-1} w^\top (J_{\text{thermal},h}, J_{\text{voltage},h}, J_{\text{econ},h}), \tag{95}$$

subject to the AC equations, all equipment bounds, and $|P_{G,i,h} - P_{G,i,h-1}| \le R_i$, with the same fixed weight $w \in \Delta^2$ across the horizon. The first ramp constraint uses the actual preceding dispatch. Reference targets are retained along complete successful chains; if any solve fails, the entire ramp-coupled chain is discarded. A one-action decoder is a special case of the response representation; the evaluation still closes the loop across the full 300-step load trajectory.

J.3 Trajectory-Level Parameter Variation

Nominal reference generation uses Latin hypercube sampling for 5000 load scenarios over $[0.6, 1.4]$ times nominal demand and 50 scalarization weights, producing 250,000 reported targets. Parameter-varied reference chains use the same ramp-coupled construction while holding one sampled branch realization fixed throughout each chain.

For each of 41 branches, draw $\widetilde{\xi}_\ell \sim \mathcal{N}(0, 0.01^2)$, set $\xi_\ell = \text{clip}(\widetilde{\xi}_\ell, -0.03, 0.03)$, and apply $\kappa_\ell = 1 + \xi_\ell$. The network is rebuilt from these parameters, and the realization remains fixed and visible to GPC throughout the trajectory. Loads vary while generator ratings and objective coefficients remain nominal, isolating branch-admittance variation. The physical dispatch interval is 5 s, distinct from decision latency.

GPC's training family includes these perturbations and its input includes the sampled descriptor. MPC uses the perturbed network model. TD3 and CPO are trained on the nominal network and evaluated without fine-tuning. Under the full evaluation constraints, neither produces a feasible dispatch in the nominal or the perturbed run (Table 2), so their failure is not attributable to the parameter change alone. The held-out realization tests whether GPC's learned map can serve a new member of its modeled parameter family without retraining; the RL results describe nominal-trained policies deployed under the same parameter change. Algorithm-specific robustness under matched perturbation training is a separate question.

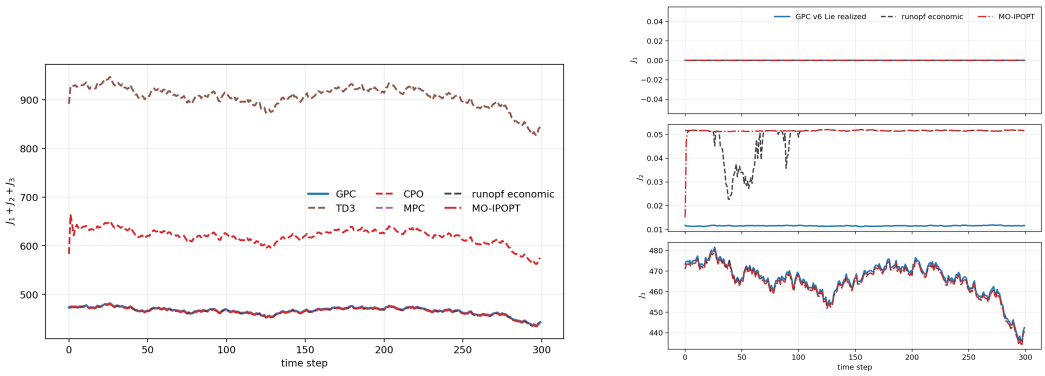
J.4 Complete Nominal and Perturbed Aggregates

Table 6: OPF cost aggregation. Cumulative gap is the method cost minus the MO-IPOPT reference cost over finite-cost steps. Constraint feasibility uses the complete 300-step diagnostic.

Network	Method	Finite costs	Cumulative cost	Cumulative MO-IPOPT gap
Nominal	GPC	300	108324.44	325.19
	MPC	300	108296.42	297.17
	CPO	300	155934.78	47935.53
	TD3	300	223129.70	115130.45
Perturbed	GPC	299	88293.29	220.02
	MPC	299	88274.32	201.05
	CPO	299	114627.72	26554.45
	TD3	299	168947.00	80873.73

The nominal GPC mean cost is 361.08 versus 360.99 for MPC, with 100% feasibility for both. The perturbed means are 295.30 and 295.23 over 299 finite costs; both have 99.67% full-constraint feasibility. The generator-limit diagnostic accounts for the one infeasible step. Cost validity, power-flow convergence, grid constraints, and generator feasibility are distinct diagnostics and must not be merged into a single claim of perfect feasibility. Mean decision times are 12.3 versus 1193.8 ms nominally and 12.6 versus 1228.7 ms under perturbation for GPC and MPC, respectively.

OPF latent-update variants. An auxiliary IEEE 30-bus check compares RK2 and RK2 with Lie-local correction on a common 300-step load trajectory, distinct from the main benchmark trajectory, and five saved response maps. All 10 controller/map runs attain 100% AC convergence and recorded constraint feasibility, with zero recorded voltage and generator violations. For the latest rollout-weighted map in this auxiliary set, the mean total cost decreases from 282.57 with RK2 to 282.00 with RK2 plus Lie-local correction. The Lie-local term modifies the RK2 latent code at all 300 steps, with a mean ℓ_2 difference of 3.56×10^{-5}, while retaining 300/300 feasible AC-converged dispatches and zero recorded thermal, voltage, generator, and slack violations. Moreover, the maximum AC mismatch is reduced from below 10^{-5} with RK2 to below 10^{-6} with the Lie-local correction. These results indicate that the Lie-local term provides a distinct geometric refinement of the RK2 update, improving AC consistency without sacrificing the observed feasibility rate and, in this auxiliary setting, also yielding a modest reduction in operating cost. These auxiliary runs use response checkpoints distinct from those in the main OPF benchmark above.



(a) Nominal total objective

(b) Objective channels

Figure 6: Nominal IEEE 30-bus trajectory plots against the multi-objective IPOPT and economic-dispatch references. GPC and MPC have 100% feasibility in the nominal benchmark summary.

J.5 Held-Out $N-k$ Topology Shift on IEEE 118-Bus

The 30-bus test varies branch parameters on a fixed graph; here we test discrete line-removal transfer on the IEEE 118-bus system with 54 generators. The offline set contains 1,001 training graphs spanning $N-0$ through $N-5$, with 50 validation and 50 test graphs held out. We report one residual-penalty GPC checkpoint, without a distributed-training or checkpoint-version comparison.

Test graph 1091 remains connected after removing branches 12, 24, 51, 105, and 179. The 60-step evaluation uses seed 42, a 5% ramp limit, the dynamic loads and priorities in Section H, sparse Newton power flow, and at most two PV-to-PQ correction rounds. The paired comparator is nonlinear AC multi-objective OPF solved with IPOPT; the gap is $100|J_{\text{GPC}} - J_{\text{MO-OPF}}|/|J_{\text{MO-OPF}}|$.

GPC is feasible at all 60 saved steps under the post-correction grid and generator diagnostics; its mean and maximum gaps are 0.1002% and 0.1335%. Excluding first-step initialization, mean time over steps 2–60 is 52.48 ms (3.03 ms decision and 49.45 ms correction) versus 2576.23 ms for paired IPOPT, a $49.1\times$ speedup. Figure 7 shows the recorded trajectory.

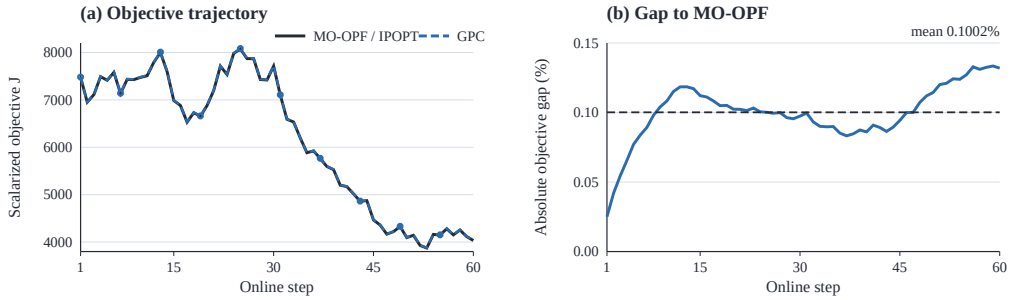


Figure 7: GPC on held-out IEEE 118-bus $N-5$ graph 1091. Left: realized objective and paired MO-OPF/IPOPT reference. Right: absolute relative gap; dashed line, 0.1002% mean. Both panels use the unchanged saved 60-step data.

This result remains tolerance-dependent: mean Newton residual is 5.47×10^{-7}, but the stricter reconstructed-balance flag is 0% (maximum mismatch 2.58×10^{-5}), so the saved feasibility diagnostic does not certify exact AC equality. One graph and one seed cannot establish test-split reliability; the conclusion is limited to response-map reuse on this recorded topology shift.

J.6 Repeated OPF Evaluations and Scope

The OPF figures show one representative trajectory per setting. To assess variability across load conditions, we repeat the IEEE 30-bus GPC evaluation on 100 different load sequences for each setting and compute GPC's gap relative to the MO-IPOPT reference for every run. Table 7 summarizes the 100 runs per setting.

Table 7: Repeated IEEE 30-bus GPC evaluations on 100 different load sequences per setting. Gaps are relative to the MO-IPOPT reference.

Setting	Runs	Median gap (%)	Mean gap (%)	Max gap (%)
Nominal	100	0.2923	0.3152	0.4262
Branch-perturbed	100	0.3238	0.3812	0.4913

The four-method comparison in Table 2 and Table 6 uses one nominal and one branch-perturbed 300-step trajectory; their reference-relative gaps of approximately 0.30% and 0.25% are of the same magnitude as the repeated-run statistics. For IEEE 118-bus, the held-out test split contains 50 independent topologies covering $N-1$ through $N-5$ (5, 10, 12, 13, and 10 graphs, respectively); the detailed online comparison saved for this study is one 60-step trajectory on test graph 1091 (Section J.5).

K Analytical Dynamics and Full Comparison

The analytical domain isolates dynamic coupling and response geometry in a four-dimensional physical state with two-dimensional acceleration. Both objective variants retain the same moving obstacle, input limits, slew constraint, and observation structure. Only the objective changes, allowing comparison of response approximation under convex and nonconvex objectives without changing the physical constraint set.

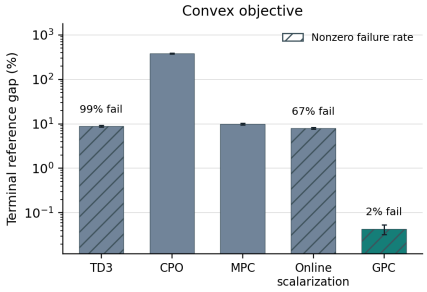
K.1 Dynamics and Objective Landscapes

For $q_t, v_t, u_t \in \mathbb{R}^2$, the double integrator follows

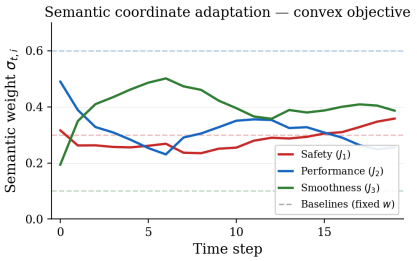
$$q_{t+1} = q_t + \Delta t\, v_t + \tfrac{1}{2}\Delta t^2 u_t, \qquad v_{t+1} = v_t + \Delta t\, u_t. \tag{96}$$

Input and slew bounds constrain $\|u_t\|$ and $\|u_t - u_{t-1}\|$. A moving elliptical keep-out region imposes

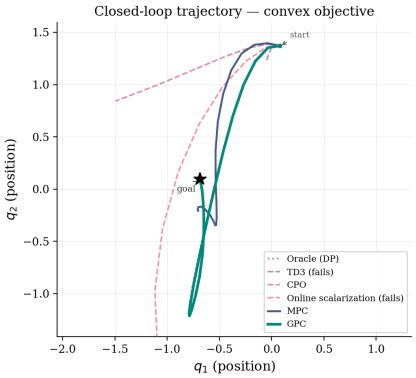
$$1 - \frac{(q_{1,t+1} - c_1(p_t))^2}{a^2} - \frac{(q_{2,t+1} - c_2(p_t))^2}{b^2} \le 0. \tag{97}$$



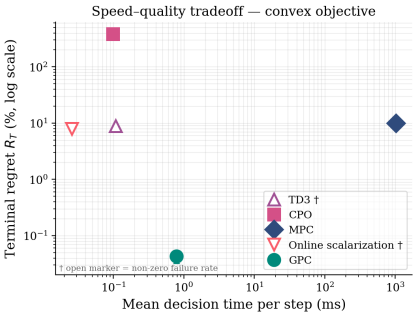
(a) Convex: reference gap



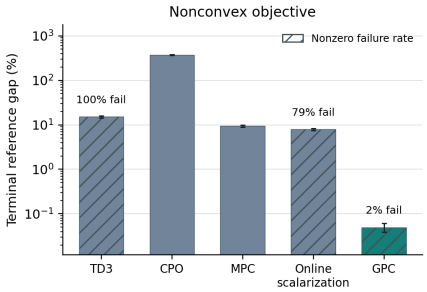
(b) Convex: priorities



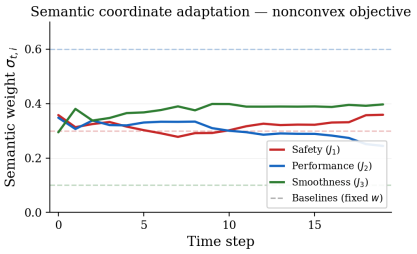
(c) Convex: trajectory



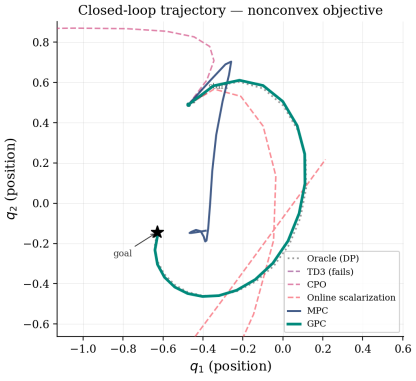
(d) Convex: computation



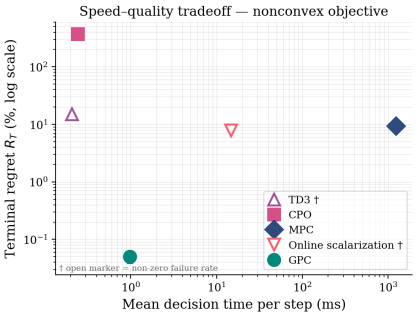
(e) Nonconvex: reference gap



(f) Nonconvex: priorities



(g) Nonconvex: trajectory



(h) Nonconvex: computation

Figure 8: Analytical benchmarks. The convex case (a–d) and nonconvex case (e–h) retain the same physical constraints. Panels (a,e) plot the dynamic-programming reference gap on a logarithmic axis; hatching denotes a nonzero trajectory failure rate. Panels (c,g) show representative closed-loop trajectories. Table 8 reports the complete gap, violation, failure, and runtime statistics.

The safety and performance objectives are

$$J_1 = \|q_{t+1} - q_{\text{safe}}(p_t)\|^2, \qquad J_2 = \|q_{t+1} - q_{\text{goal}}\|^2 + \beta_u \|u_t\|^2. \tag{98}$$

The nonconvex variant adds $\lambda_{\text{nc}} \sum_{\ell=1}^{2} [1 - \cos(\omega(q_{\ell,t+1} - q_{\text{goal},\ell}))]$ to J_2. The convex designation concerns the objective, not the obstacle-constrained feasible set. The discretized finite-horizon dynamic-programming reference uses the same model and constraints; its finite grid defines the comparator accuracy.

K.2 Reference Construction and Baselines

For each context $c_t = (q_t, v_t, u_{t-1}, p_t)$, the target family varies the safety/performance scalarization weight while propagating the same double-integrator dynamics. A finite-horizon dynamic-programming reference discretizes state and control and applies the corresponding constrained Bellman recursion. It is an evaluation reference in this low-dimensional setting; no comparable grid-based reference is assumed available for multi-agent navigation or OPF.

The baselines include TD3, CPO, MPC, and online scalarization. Online scalarization optimizes the current weighted decision and can therefore have low per-decision cost while failing the longer dynamically coupled task. MPC uses a receding-horizon physics model. CPO and TD3 use their trained policy evaluations. The reported comparison separates reference gap, step-level constraint violations, episode-level failure, and runtime so that a good value of one metric does not conceal a failure in another.

K.3 Metric Definitions and Additional Diagnostics

The terminal metric is a normalized accumulated reference gap,

$$R_T = \frac{\sum_{t=1}^{T} \sigma_t^\top [\mathbf{J}(s_t, u_t) - \mathbf{J}^{\text{DP}}(s_t)]}{\sum_{t=1}^{T} \sigma_t^\top \mathbf{J}^{\text{DP}}(s_t) + \delta_R}, \qquad \delta_R > 0. \tag{99}$$

The table reports $100 R_T$ as a percentage. This empirical metric is separate from the matched-action theorem.

Table 8: 100-trajectory analytical summaries. Gaps are percentages, mean $\pm$ s.e.m. Step-level violation fractions and trajectory failure rates are reported separately.

Objective	Method	Gap (%)	Obs. viol.	Slew viol.	Fail (%)	Time (ms)
Convex	TD3	8.95 ± 0.40	0.08	52.49	99	0.11
	CPO	382.55 ± 9.91	0.00	0.00	0	0.10
	MPC	9.93 ± 0.51	0.00	0.00	0	1025.24
	Online scalarization	7.90 ± 0.35	0.00	0.00	67	0.03
	GPC	0.043 ± 0.011	0.00	0.00	2	0.78
Nonconvex	TD3	15.11 ± 0.68	0.07	75.63	100	0.21
	CPO	370.63 ± 9.59	0.00	0.00	0	0.24
	MPC	9.43 ± 0.52	0.00	0.00	0	1229.62
	Online scalarization	7.90 ± 0.38	0.00	0.00	79	14.65
	GPC	0.050 ± 0.011	0.05	0.00	2	0.97

L Discussion: What Is Amortized and What Remains Online

GPC amortizes constrained response construction by organizing solutions across contexts and trade-offs offline, then adjusting a localized response online. This is useful when many decisions reuse related geometry; disconnected solution branches can instead make dataset coverage and representation capacity limiting. The navigation results evaluate the response-map framework with learned plan correction and finite refinement; the explicit RK2/Lie update belongs to the latent-flow instantiation used for power-flow optimization.

Physical indicators determine which objective receives emphasis, but their units, normalization, filtering, and step size still require calibration. Finite evaluations establish observed outcomes, the

tube certificate additionally requires uniform regularity and coverage, and stability or global optimality requires stronger assumptions. Matched-budget refinement, matched-information policies, broader operating envelopes, and certified decoder coverage would test finer component-level and out-of-family claims.