Infeasible optimization problems and the hierarchical augmented Lagrangian method in imitation learning

Roland Andrews, Justin Carpentier, and Ajay Sathya

***Abstract*— Imitation learning (IL) is an effective approach to train complex robotics policies. Recent works have introduced hard constraints into imitation-learning optimization problems to ensure safety, stability, and robustness of the learned policy. However, we argue that these constraints are sometimes infeasible, which can lead to unstable or difficult training dynamics. We study a simple remedy for such situations based on recent theoretical results on the augmented Lagrangian method in infeasible settings. We show that our approach drives the learned policy toward the solution of a closest-feasible constrained IL problem with desirable properties. The method is illustrated on a toy driving example with a total-acceleration constraint and pedestrian-safety constraints, a setting in which infeasibility can naturally arise while still allowing a safe learned policy.**

I. INTRODUCTION

Imitation learning (IL) is an effective approach for training complex robotic policies: it uses expert demonstrations to define all or part of the loss function used to train a policy. IL is particularly effective when a loss function cannot otherwise be specified easily, or when an available loss provides only a weak gradient signal over the relevant regions of the state space. Behavioral cloning is a common IL methodology that consists in learning to imitate expert actions $U^* = (u_k^*)_{0 \leq k \leq K}$ together with their corresponding states $\Xi^* = (\xi_k)_{0 \leq k \leq K}$:

$$\underset{\hat{u} \in \mathcal{D}}{\text{minimize}} \quad \mathcal{L}_{U^*, \Xi^*}(\hat{u}) + r(\hat{u}) \tag{1}$$

Here $\mathcal{D}$ is a subset of a Hilbert space of functions contained in L^2, typically parameterized by a function class such as a family of neural networks, and $\mathcal{L}$ is a loss that encourages the learned policy to behave similarly to the expert demonstration. r is a regularizer that prevents overfitting; for the theoretical analysis we assume that r is convex in $\hat{u}$ and level bounded. In practice, r is often not convex in $\hat{u}$. For instance, if $\hat{u}$ is represented by a neural network and r is a squared penalty on its weights, as in weight decay, then r is convex in the parameters of the neural network, but not on the space of neural networks. We nevertheless observe in Section III that the conclusions suggested by the convex analysis remain informative in practice. In the present work we illustrate our results by choosing the loss $\mathcal{L}_{U^*, \Xi^*}(\hat{u}) = \frac{1}{K} \sum_{k=0}^{K} \|\hat{u}(\xi_k) - u_k^*\|^2$. This type of behavioral cloning typically suffers from compounding errors along the learned policy's trajectory, but the results of the present work can easily be extended to more elaborate loss functions.

Optimization problem (1) suffers from safety, robustness, and stability issues. Indeed, the expert behavior may itself violate constraints that we would like the learned policy to satisfy (for instance, a human driver who sometimes exceeds the speed limit or fails to remain perfectly within lane boundaries). It may also happen that the expert data are themselves safe and robust, but that an embodiment mismatch, such as differences in morphology, actuation, or dynamics, makes naive imitation unsafe. One may think, for example, of a robot learning to walk from human walking data. For these reasons, the recent literature has studied constrained imitation learning:

$$\begin{aligned} \underset{\hat{u} \in \mathcal{D}}{\text{minimize}} \quad & \mathcal{L}_{U^*, \Xi^*}(\hat{u}) + r(\hat{u}) \\ \text{s.t.} \quad & C(\hat{u}) \in \mathcal{K} \end{aligned} \tag{2}$$

The condition $C(\hat{u}) \in \mathcal{K}$ represents constraints added to ensure that the learned policy is safe, stable, and robust.

For the theoretical discussion we assume that $\mathcal{D}$ is a convex and closed. We also assume that the mapping C and the convex cone $\mathcal{K}$ are such that the set

$$\{(\hat{u}, \phi) \mid \phi \in C(\hat{u}) - \mathcal{K}\}$$

is convex, closed, and non-empty. When problem (2) is feasible, this last condition simply corresponds to stating that the constraints are convex. We denote by $\mathcal{K}^\circ$ the polar cone of $\mathcal{K}$. These assumptions guarantee that (2) is a proper, closed, convex problem. Of course the optimization problem is generally non-convex in the parameters of the function class that parameterizes $\hat{u}$.

Infeasibility of (2) arises when the set $\{\hat{u} \mid C(\hat{u}) \in \mathcal{K}\}$ is empty. We argue in the literature review of Section I-A, as well as through the simple yet intuitive example of Section III, that infeasibility may naturally arise in many constrained IL settings.

A. Related Works

The existing literature justifies the constrained imitation learning setting (2). However, the cited works do not all formulate constrained IL in the same way: some constrain imitation through reward lower bounds, some through Lyapunov or semidefinite certificates, some through control barrier functions, and some through differentiable completion and correction procedures. Infeasibility is therefore a plausible issue in several of these methods, but it is usually not the main object of study.

[1] introduces Reward-Constrained Behavior Cloning, which combines a behavior-cloning objective with a lower-bound constraint on expected return, or equivalently on an expected Q-value under the learned policy. The goal is not

merely to imitate the demonstration, but to preserve desirable behavior patterns from demonstrations while avoiding severe performance degradation. This paper is directly relevant to infeasibility, since the reward threshold is user-chosen and the authors explicitly note that "the constraints may be unsatisfied for a long time, which [...] makes the learning process unstable". The feasibility of the constrained problem depends critically on the chosen lower bound as well as on the constantly evolving estimated Q-value.

[2] proposes a kernelized imitation learning framework for trajectory generation with obstacle avoidance and linear or nonlinear hard constraints. Their method relies on local linearization of nonlinear costs and constraints together with iterative constrained updates using Lagrange multipliers and proximal regularization. Interestingly, their setting matches exactly the convex setting (2).

[3] derives convex stability and safety conditions for certain neural-network controllers in linear time-invariant systems by combining Lyapunov arguments with quadratic constraints on the nonlinearities. The resulting constrained imitation learning problem is solved with an ADMM-based algorithm. Since ADMM can be interpreted as a splitting scheme related to augmented Lagrangian methods, it is likely that our results may be extendable to their algorithm and ADMM in general, although we do not investigate that question here.

In [4], Control Barrier Functions are used to transfer safety guarantees from a robust expert controller to an end-to-end imitation-learned controller. Their guarantees are formulated in terms of set invariance and input-to-state safety, and the robust expert controller is obtained through a Tunable Robust Optimization Program (TR-OP), which is a constrained imitation-learning optimization problem that accounts for matched disturbances and state uncertainty. When such safety constraints are combined with additional design requirements, infeasibility or excessive conservatism may arise in practice. Although we do not use their TR-OP, we illustrate a case of infeasibility that would apply to their examples in Section III. A related direction is [5], where a customized control barrier function is itself learned from safety demonstrations and then used for shielding in a learning-from-demonstration pipeline. It is possible that learning the constraints may itself increase the chance of infeasible constraints.

[6] proposes a Differentiable Constrained Imitation Learning framework combining three ingredients: a neural network predicts controls, an explicit completion step enforces equality constraints induced by the dynamics, and a gradient-based correction step reduces inequality violations. During training, soft penalty terms are also used on the remaining constraint residuals. Thus, their method is more structured than a plain penalty method. Importantly for our purposes, they explicitly discuss robustness to incorrect constraints that may render the problem infeasible. This is close in spirit to our motivation, although their work does not provide a general convergence theory for infeasible constrained IL problems.

B. Contribution

Our contribution is to analyze the issue of infeasibility explicitly and to provide a principled behavior when the imposed constrained IL problem is infeasible. We build on existing work on the inexact augmented Lagrangian method (IALM) and extend the theory to the case of an augmented Lagrangian method with several distinct penalty terms instead of a single uniform penalty for all constraints. We show that in this setting, the IALM converges to a well-defined analytical "closest feasible problem" (7) with desirable properties. Namely, it respects the most important constraints while minimizing the violation of the constraints of lesser importance. We illustrate the theory with a toy experiment.

II. The IALM algorithm for safe infeasible IL

The behavior of the augmented Lagrangian method (ALM) for infeasible problems was first studied for quadratic programming (QP) in [7]. Recent works [8], [9] study the general convex case. They show that when the constraints of a convex problem are infeasible, the ALM converges to the "closest feasible problem," defined as the problem with the same objective and constraints shifted by the smallest quantity that makes them feasible. Equivalently, it is the problem with the same objective in which the constraint violation norm is minimized first (see [9, Equation 34] for the analytical formulation).

In our setting, we separate the constraints into two types with different penalty sequences, and we show that this allows us to choose which constraints should be satisfied exactly and which should only have their violation minimized. This is an improvement, demonstrated here in the present setting, over the framework of [8], [9] where all constraints have the same penalty term.

We rewrite (2) by separating the constraints into two groups:

$$\begin{aligned} \underset{\hat{u}\in\mathcal{D}}{\text{minimize}} \quad & \mathcal{L}_{U^*,\Xi^*}(\hat{u}) + r(\hat{u}) \\ \text{s.t.} \quad & C_A(\hat{u}) \in \mathcal{K}_A \\ & C_B(\hat{u}) \in \mathcal{K}_B. \end{aligned} \tag{3}$$

We use the notation $\lambda = (\lambda_A, \lambda_B)$, $\mathcal{K} = \mathcal{K}_A \times \mathcal{K}_B$, and $C(\cdot) = (C_A(\cdot), C_B(\cdot))$.

The augmented Lagrangian (dropping constant terms, see [9, Appendix A] for details on reformulations and variants of the augmented Lagrangian) is

$$\begin{aligned} L_{\gamma_A,\gamma_B}(\hat{u},\lambda_A,\lambda_B) = f(\hat{u}) &+ \frac{\gamma_A}{2}\left\|\mathrm{Proj}_{\mathcal{K}_A^\circ}\left(C_A(\hat{u}) - \frac{\lambda_A}{\gamma_A}\right)\right\|^2 \\ &+ \frac{\gamma_B}{2}\left\|\mathrm{Proj}_{\mathcal{K}_B^\circ}\left(C_B(\hat{u}) - \frac{\lambda_B}{\gamma_B}\right)\right\|^2 \end{aligned} \tag{4}$$

We write $u \underset{\varepsilon}{\approx} \underset{\hat{u}\in\mathcal{D}}{\operatorname{argmin}} L_{\gamma_A,\gamma_B}(\hat{u},\lambda_A,\lambda_B)$ to mean that u approximately minimizes $L_{\gamma_A,\gamma_B}(\cdot,\lambda_A,\lambda_B)$ with error ε in

Algorithm 1 Hierarchical IALM

Require: Problem data, initial $\hat{u}^0$, λ^0, sequence of positive errors $(\varepsilon_k)_{i\in\mathbb{N}}$, sequences of positive penalties $(\gamma_{A,i}, \gamma_{B,i})_{i\in\mathbb{N}}$

1: **for** $i = 0, 1, \ldots$ **do**
2: $\quad \hat{u}^{i+1} \underset{\varepsilon_k}{\approx} \arg\min_{\hat{u}\in\mathcal{D}} \; L_{\gamma_{A,i},\gamma_{B,i}}(\hat{u}, \lambda_A^i, \lambda_B^i)$
3: $\quad \lambda_A^{i+1} \leftarrow -\gamma_{A,i}\mathrm{Proj}_{\mathcal{K}_A^\circ}(C(\hat{u}^{i+1}) - \lambda_A^i/\gamma_{A,i})$
4: $\quad \lambda_B^{i+1} \leftarrow -\gamma_{B,i}\mathrm{Proj}_{\mathcal{K}_B^\circ}(C(\hat{u}^{i+1}) - \lambda_B^i/\gamma_{B,i})$
5: **end for**

the sense that

$$L_{\gamma_A,\gamma_B}(u, \lambda_A, \lambda_B) - \operatorname*{argmin}_{\hat{u}\in\mathcal{D}} L_{\gamma_A,\gamma_B}(\hat{u}, \lambda_A, \lambda_B) \le \varepsilon \quad (5)$$

Hypothesis 1: The step sizes $(\gamma_k)_{k\in\mathbb{N}}$ are positive, the errors $(\varepsilon_k)_{k\in\mathbb{N}}$ are nonnegative, and the following assumptions hold:

- $\sum_{k=1}^{\infty} \varepsilon_k < \infty$.
- $\frac{\varepsilon_{k+1}}{\gamma_k} \sum_{i=0}^{k} \gamma_i = O\left(\frac{1}{\sqrt{\sum_{i=0}^{k} \gamma_i}}\right)$.
- $\sum_{k=1}^{\infty} \left((\varepsilon_k/\gamma_k)^2 \sum_{i=0}^{k-1} \gamma_i\right) < \infty$.
- $\sum_{k=1}^{\infty} \left((\varepsilon_k/\gamma_k) \sum_{i=1}^{k} \gamma_{i-1}\right) < \infty$.
- $\sum_{k=0}^{\infty} \gamma_k = \infty$.

These assumptions ensure that the error decreases sufficiently fast to zero. For example, if the step sizes are constant $\gamma_i = \gamma$, then the choice of error $\varepsilon_k = \frac{1}{((k+1)\ln(k+1))^2}$ satisfies all the assumptions. The last hypothesis $\sum_{k=0}^{\infty} \gamma_k = \infty$ is standard in the optimization literature.

Theorem 2: Assume that the sequences $(\gamma_{A,i}, \gamma_{B,i})_{i\in\mathbb{N}}$ and $(\varepsilon_k)_{i\in\mathbb{N}}$ are such that Hypothesis 1 is satisfied for both sequences of step sizes, and that $\gamma_{A,i}/\gamma_{B,i} \to 0$. Assume moreover that $C_B(\hat{u}) \in \mathcal{K}_B$ is feasible and that the constraints have no recession direction as defined in [9, Definition 3]. Then the iterates of Algorithm 1 converge to the solutions of the trilevel problem

$$\begin{aligned} \underset{\hat{u}\in\mathcal{D}}{\text{minimize}} \quad & \mathcal{L}_{U^*,\Xi^*}(\hat{u}) + r(\hat{u}) \quad (6) \\ \text{s.t.} \quad & \hat{u} \in \operatorname*{argmin}_{\tilde{u}\in\mathcal{D}} \quad \left\|C_A(\tilde{u}) - \mathrm{Proj}_{\mathcal{K}_A}(C_A(\tilde{u}))\right\| \\ & \text{s.t.} \quad C_B(\hat{u}) \in \mathcal{K}_B. \end{aligned}$$

Proof: The proof is given in Appendix I. ∎

Remark 3: When both constraints $C_A(\hat{u}) \in \mathcal{K}_A$ and $C_B(\hat{u}) \in \mathcal{K}_B$ are feasible simultaneously, (7) is equal to (3). Indeed, in that case, $\min_{\tilde{u}\in\mathcal{D}} \quad \left\|C_A(\tilde{u}) - \mathrm{Proj}_{\mathcal{K}_A}(C_A(\tilde{u}))\right\| = 0$ simply means $C_A(\hat{u}) \in \mathcal{K}_A$. Thus, when problem (3) is feasible, Algorithm 1 simply solves it.

III. Toy-car experiment

In autonomous driving, one may wish to impose both a maximum admissible acceleration and a constraint ensuring a safe distance from pedestrians. We call constraint A the constraint related to maximum acceleration and constraint B the one related to pedestrian safety. Such constraints allow one to learn from expert examples while favoring safe behavior. However, a demonstration in which the driver brakes sharply to avoid a collision may be infeasible: respecting the maximum acceleration constraints forbids braking sharply, and respecting a safety constraint around pedestrians might require sharp braking, and therefore high acceleration, if the vehicle has significant speed and the pedestrian is detected late. Discarding such data would not be a good solution, since they still contain expert behavior in critical scenarios; indeed, even an autonomous vehicle might encounter pedestrians who step onto the road unexpectedly.

We study a toy experiment in which a car, modeled as a kinematic bicycle, drives on a closed stadium-shaped track and a pedestrian may appear suddenly on a crosswalk. See Figure 1 for a representation of the setting. The vehicle state is $x = (p_x, p_y, \psi, v)$, where (p_x, p_y) is the position of the vehicle, ψ its heading, and v its longitudinal speed. The control is $u = (a, \tau)$, where a is the longitudinal acceleration and $\tau = \tan(\delta)$ is the steering variable associated with the front-wheel angle δ. Expert demonstrations are generated by a pure-pursuit controller perturbed by Ornstein–Uhlenbeck noise in both control channels. When a pedestrian appears on the road at short distance, the expert switches to a braking mode with constant negative acceleration until the vehicle comes to a stop. The braking level is randomized, and not all expert demonstrations successfully maintain a safe distance from the pedestrian (see Table I).

The learned policy is a neural network that receives the vehicle state together with a one-hot pedestrian-visibility input. In our experiments, the lower-priority constraint A is a total-acceleration bound, while the higher-priority constraint B is a pedestrian-braking constraint that is active only when the pedestrian is visible. Additional experimental details are given in Appendix II.

More precisely, in the experiment we use a quadratic constraint on the controls $(\hat{a}_k, \hat{\tau}_k)$:

$$C_A(\hat{u}) \in \mathcal{K}_A \quad \Longleftrightarrow \quad \forall k,\ \hat{a}_k^2 + \left(\frac{v_k^2}{L}\hat{\tau}_k\right)^2 \le a_{\mathrm{tot,max}}^2,$$

where L denotes the wheelbase of the kinematic bicycle model. Moreover,

$$C_B(\hat{u}) \in \mathcal{K}_B \quad \Longleftrightarrow \quad \forall k_p,\ \hat{a}_{k_p} \le -\frac{v_{k_p}^2}{2d_{\mathrm{ped},k_p}},$$

where k_p indexes the data points at which a pedestrian is visible. $d_{\mathrm{ped},k}$ denotes the free distance from the front of the vehicle to the pedestrian, including a safety margin.

We compare Algorithm 1 with two natural alternatives. The three methods are:

- Hierarchical IALM: Algorithm 1 with $\gamma_{A,i} = 5/(i+1)$ and $\gamma_{B,i} = 15$
- Only total-acceleration constraint: the pedestrian-braking constraint is ignored, and only the total-acceleration constraint is kept, with $\gamma_{A,i} = 5/(i+1)$

TABLE I

COMPARISON OF PEDESTRIAN COLLISION RATES (%) UNDER DIFFERENT CONTROL/OPTIMIZATION SCHEMES.

Method	Expert Controller	Hierarchical IALM	Only Total Acceleration Constraint	Both Constraints at Same Level
Collision Rate (%)	18%	0%	31%	14%

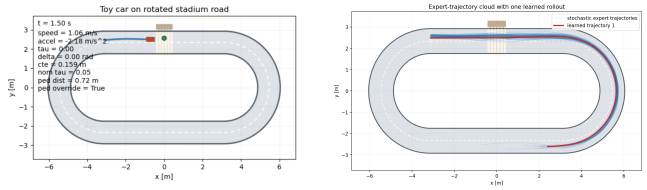
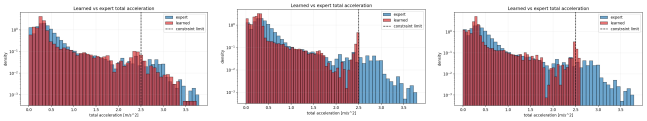


Fig. 1. Left: overview of the track together with the pedestrian (green disc) at the moment it appears on the road and an expert car (red rectangle) rollout. Right: cloud of expert trajectories (blue) together with one learned rollout (red), illustrating the path of the training data.



(a) Hierarchical IALM (b) Total-acceleration constraint only (c) Both constraints at the same level

Fig. 2. Comparison of the learned and expert total-acceleration distributions under the three optimization schemes considered in the toy-car study. y-axis is log-scaled.

- Both constraints at the same level: the ALM is used with $\gamma_{A,i} = \gamma_{B,i} = 15$. The two constraints are treated equally.

The results in Figure 2 indicate that Hierarchical IALM correctly prioritizes the pedestrian-safety constraint, yielding no collisions while minimizing the violation of the lower-priority constraint. This is consistent with the theory of Theorem 2. By comparison, when only the acceleration constraint is kept and no pedestrian-safety constraint is imposed, the learned policy respects that constraint but produces more pedestrian collisions than the expert policy on which it was trained, since it cannot decelerate hard enough to avoid collision. Finally, if one simply applies the usual IALM with identical penalty terms on all constraints, as in [9], the result is a compromise in which neither constraint is respected perfectly, but the learned policy violates both constraints less than the expert. This is also consistent with the theory in [9, Theorem 3.5].

IV. CONCLUSION

In this work, we study constrained imitation-learning problems in the infeasible case, when safety, robustness, and stability constraints may conflict with one another. We argue that such infeasible settings arise in practice. We propose a two-penalty inexact augmented-Lagrangian method that provides guarantees on the behavior of the imitation-learning optimization problem and an analytically tractable limit for the learned policy. We show that, in the convex setting, the method converges to a lexicographic closest-feasible problem: the high-priority constraints are enforced first when feasible, the lower-priority constraint violations are then minimized, and the imitation objective is optimized over that set.

The toy-car study illustrates this mechanism on a simple but representative example, where pedestrian safety and total-acceleration limits can conflict on emergency-braking demonstrations. In that setting, the trilevel hierarchy produces safer behavior than alternatives while remaining close to the expert distribution. These results suggest that infeasibility can be treated effectively in constrained imitation learning, and that hierarchical augmented-Lagrangian formulations are a promising way to do so. A natural next step is to extend the analysis to richer nonconvex policy classes and more realistic large-scale benchmarks.

REFERENCES

[1] Z. Wang, M. Wang, J. Zhang, Y. Chen, C. Zhang, and Z.-H. Zhou, "Reward-constrained behavior cloning." in *IJCAI*, 2021, pp. 3169–3175.

[2] Y. Huang, "Ekmp: Generalized imitation learning with adaptation, nonlinear hard constraints and obstacle avoidance," *arXiv preprint arXiv:2103.00452*, 2021.

[3] H. Yin, P. Seiler, M. Jin, and M. Arcak, "Imitation learning with stability and safety guarantees," *IEEE Control Systems Letters*, vol. 6, pp. 409–414, 2021.

[4] R. K. Cosner, Y. Yue, and A. D. Ames, "End-to-end imitation learning with safety guarantees using control barrier functions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 5316–5322.

[5] Y. Yang, L. Chen, Z. Zaidi, S. van Waveren, A. Krishna, and M. Gombolay, "Enhancing safety in learning from demonstration algorithms via control barrier function shielding," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 820–829.

[6] C. Diehl, J. Adamek, M. Krüger, F. Hoffmann, and T. Bertram, "Differentiable constrained imitation learning for robot motion planning and control," *arXiv preprint arXiv:2210.11796*, 2022.

[7] A. Chiche and J. C. Gilbert, "How the augmented lagrangian algorithm can deal with an infeasible convex quadratic optimization problem," *Journal of Convex Analysis*, vol. 23, no. 2, 2016.

[8] Y.-H. Dai and L. Zhang, "The augmented lagrangian method can approximately solve convex optimization with least constraint violation," *Mathematical Programming*, vol. 200, no. 2, pp. 633–667, 2023.

[9] R. Andrews, J. Carpentier, and A. Taylor, "Augmented lagrangian methods for infeasible convex optimization problems and diverging proximal-point algorithms," *arXiv preprint arXiv:2506.22428*, 2025.

APPENDIX I
PROOF OF THE MAIN THEOREM

Theorem 4: Assume that $(\gamma_{A,i}, \gamma_{B,i})_{i\in\mathbb{N}}$ and $(\varepsilon_k)_{i\in\mathbb{N}}$ are such that Hypothesis 1 is satisfied for both sequences of step sizes, and that $\gamma_{A,i}/\gamma_{B,i} \to 0$. Assume moreover that $C_B(\hat{u}) \in \mathcal{K}_B$ is feasible and that the constraints have no recession direction as defined in [9, Definition 3]. Then the iterates of Algorithm 1 converge to the solutions of the

trilevel problem

$$\begin{aligned} \underset{\hat{u}\in\mathcal{D}}{\text{minimize}} \quad & \mathcal{L}_{U^*,\Xi^*}(\hat{u}) + r(\hat{u}) \\ \text{s.t.} \quad & \hat{u} \in \underset{\tilde{u}\in\mathcal{D}}{\operatorname{argmin}} \quad \left\| C_A(\tilde{u}) - \operatorname{Proj}_{\mathcal{K}_A}(C_A(\tilde{u})) \right\| \\ & \text{s.t.} \quad C_B(\hat{u}) \in \mathcal{K}_B. \end{aligned} \tag{7}$$

Proof: The proof is a weighted version of the infeasible augmented-Lagrangian argument of [9]. The only additional point is that the two constraint blocks are penalized with different sequences, and the ratio

$$\frac{\gamma_{A,i}}{\gamma_{B,i}} \to 0$$

induces a lexicographic limit. In this appendix we also make explicit one assumption that is implicit in the statement of the theorem: to obtain exactly the condition $C_B(\hat{u}) \in \mathcal{K}_B$ in the limit problem, we assume that the set

$$\{\hat{u} \in \mathcal{D} \mid C_B(\hat{u}) \in \mathcal{K}_B\}$$

is nonempty. Otherwise, one would first obtain a closest-feasible problem for the B-block.

a) Step 1: residual formulation.: Introduce residuals

$$s_A = C_A(\hat{u}) - y_A, \qquad s_B = C_B(\hat{u}) - y_B,$$

with $y_A \in \mathcal{K}_A$ and $y_B \in \mathcal{K}_B$, and define the attainable residual set

$$\mathcal{S} \triangleq \{(s_A, s_B) \mid \exists \hat{u} \in \mathcal{D},\ s_A \in C_A(\hat{u}) - \mathcal{K}_A,\ s_B \in C_B(\hat{u}) - \mathcal{K}_B\}.$$

Because the B-constraints are feasible, the slice

$$\{s_A \mid (s_A, 0) \in \overline{\mathcal{S}}\}$$

is nonempty. We then define the lexicographically minimal residual set

$$\mathcal{S}_{\text{lex}} \triangleq \arg\min \left\{ \|s_A\| \;\middle|\; (s_A, 0) \in \overline{\mathcal{S}} \right\}.$$

Thus the trilevel problem of the theorem consists in enforcing first the exact feasibility of the B-block, then the smallest possible violation of the A-block, and finally the smallest value of the original objective $f = \mathcal{L}_{U^*,\Xi^*} + r$.

b) Step 2: weighted closest-feasible residuals.: For each iteration i, let

$$\bar{s}^i = (\bar{s}^i_A, \bar{s}^i_B) \in \arg \min_{(s_A,s_B)\in\overline{\mathcal{S}}} \left(\gamma_{A,i}\|s_A\|^2 + \gamma_{B,i}\|s_B\|^2\right).$$

Pick any $\tilde{s}_A \in \mathcal{S}_{\text{lex}}$. Since $(\tilde{s}_A, 0) \in \overline{\mathcal{S}}$, optimality of $\bar{s}^i$ gives

$$\gamma_{A,i}\|\bar{s}^i_A\|^2 + \gamma_{B,i}\|\bar{s}^i_B\|^2 \le \gamma_{A,i}\|\tilde{s}_A\|^2.$$

Dividing by $\gamma_{B,i}$ yields

$$\frac{\gamma_{A,i}}{\gamma_{B,i}}\|\bar{s}^i_A\|^2 + \|\bar{s}^i_B\|^2 \le \frac{\gamma_{A,i}}{\gamma_{B,i}}\|\tilde{s}_A\|^2.$$

Since $\gamma_{A,i}/\gamma_{B,i} \to 0$, we immediately obtain

$$\bar{s}^i_B \to 0.$$

The same inequality shows that $(\bar{s}^i_A)$ is bounded. Any accumulation point (s^∞_A, s^∞_B) of $(\bar{s}^i)$ therefore satisfies $s^\infty_B = 0$ and

$$\|s^\infty_A\| \le \|\tilde{s}_A\|.$$

By minimality of $\tilde{s}_A$, this implies $s^\infty_A \in \mathcal{S}_{\text{lex}}$. Therefore

$$\bar{s}^i_B \to 0, \qquad \operatorname{Dist}(\bar{s}^i_A, \mathcal{S}_{\text{lex}}) \to 0.$$

So the weighted closest-feasible residuals converge to the lexicographic residual set.

c) Step 3: the algorithm tracks the weighted residuals.: The proof of [9] extends to the present setting by replacing the scalar penalty parameter with the block-diagonal matrix

$$\Gamma_i \triangleq \begin{pmatrix} \gamma_{A,i} I & 0 \\ 0 & \gamma_{B,i} I \end{pmatrix}.$$

Equivalently, the dual proximal-point interpretation is unchanged, except that the Euclidean quadratic is replaced by the weighted quadratic induced by Γ_i^{-1}. Under the same assumptions on errors and stepsizes as in [9, Hypothesis 1], applied separately to $(\gamma_{A,i})$ and $(\gamma_{B,i})$, the residuals generated by the algorithm track the weighted closest-feasible vectors $\bar{s}^i$. In particular, one gets the analog of the one-penalty residual-tracking estimate

$$\|s^i_A - \bar{s}^i_A\| \to 0, \qquad \|s^i_B - \bar{s}^i_B\| \to 0,$$

and therefore

$$s^i_B \to 0, \qquad \operatorname{Dist}(s^i_A, \mathcal{S}_{\text{lex}}) \to 0.$$

This is the point at which the condition $\gamma_{A,i}/\gamma_{B,i} \to 0$ becomes decisive: the algorithm follows the weighted closest-feasible residuals, and those residuals themselves converge to a lexicographic limit.

d) Step 4: boundedness of the primal sequence.: The no-recession-direction assumption is exactly the assumption used in [9] to obtain level boundedness with respect to perturbations of the constraints. Since the residual sequence (s^i_A, s^i_B) remains eventually in a bounded neighborhood of $\mathcal{S}_{\text{lex}} \times \{0\}$, and since the same dual estimates as in [9] keep the objective values $f(\hat{u}^i)$ eventually bounded from above, the sequence $(\hat{u}^i)$ is bounded.

e) Step 5: identification of accumulation points.: Take a convergent subsequence $\hat{u}^{i_k} \to \hat{u}^\infty$. Because

$$s^{i_k}_B \to 0, \qquad \operatorname{Dist}(s^{i_k}_A, \mathcal{S}_{\text{lex}}) \to 0,$$

and because the residual graph is closed under our standing convexity and closedness assumptions, we obtain

$$C_B(\hat{u}^\infty) \in \mathcal{K}_B$$

and

$$C_A(\hat{u}^\infty) - \operatorname{Proj}_{\mathcal{K}_A}(C_A(\hat{u}^\infty)) \in \mathcal{S}_{\text{lex}}.$$

Hence $\hat{u}^\infty$ is feasible for the first two levels of the trilevel problem. It remains to identify the value of the objective on this lexicographically feasible set. This is done exactly as in the one-penalty argument of [9]: lower semicontinuity of f, together with residual convergence and boundedness, implies that every accumulation point attains the minimum value of f over the lexicographically feasible set.

f) Conclusion.: Every accumulation point of $(\hat{u}^i)$ is therefore a solution of (7). Since the sequence is bounded, if

$$\mathrm{Dist}\big(\hat{u}^i, \mathrm{Sol}((7))\big)$$

did not converge to zero, one could extract a subsequence staying at a positive distance from the solution set, and this subsequence would still admit an accumulation point. But every accumulation point belongs to the solution set, which is a contradiction. Therefore the whole sequence converges to the solution set of the trilevel problem. ■

APPENDIX II
ADDITIONAL DETAILS ON THE TOY-CAR BENCHMARK

The toy-car environment is based on a kinematic bicycle system with state (x, y, ψ, v) and controls (a, τ), where $\tau = \tan(\delta)$. The road is a closed stadium-shaped track composed of two straights and two semicircular turns.

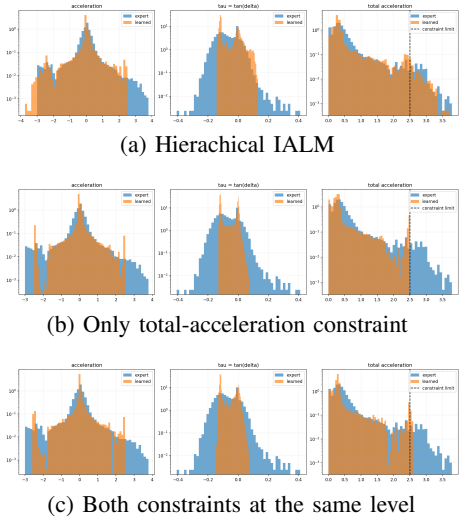
a) Expert demonstrations.: The nominal expert is a pure-pursuit controller with curvature-dependent speed tracking. Its controls are perturbed by Ornstein–Uhlenbeck noise in both acceleration and steering channels. A pedestrian event is sampled independently for each rollout. When the event is active and the car reaches the trigger region near the crosswalk, the pedestrian becomes visible for 3 seconds and the expert switches to an emergency override: the steering command is set to zero and the longitudinal acceleration is replaced by a constant sampled braking value in the interval $[-3.0, -2.0]$, until the speed drops below a small stop threshold.

b) Dataset and policy class.: The experiments use 1000 expert trajectories with time step 0.05 s and dataset horizon 16 s. Initial conditions are sampled near the outer lane, with fixed initial progress, lateral jitter 0.2, heading jitter 0.1 rad, and initial speed uniformly sampled in $[1.2, 1.9]$. The learned policy is a multilayer perceptron with hidden widths $[512, 512, 512, 512]$ and $\tanh$ activations. Its input is the normalized vehicle state together with a one-hot encoding of pedestrian visibility, and its output is the control pair (a, τ).

c) Constraints and training.: In the present experiments, the lower-priority constraint is the total-acceleration residual

$$\hat{a}^2 + \left(\frac{v^2}{L}\hat{\tau}\right)^2 - a_{\mathrm{tot,max}}^2 \le 0,$$

with $a_{\mathrm{tot,max}} = 2.5$. The higher-priority constraint is a pedestrian-braking residual based on the approximate free distance between the midpoint of the front edge of the car and the pedestrian disk, with a 10% radius margin and a clipped braking bound. This second constraint is enforced only on visible-pedestrian samples. The experiments do not include any lane-boundary CBF term or steering-centering loss. Training uses batch size 2000, 15 outer augmented-Lagrangian iterations, 5000 inner optimization steps per outer iteration, $\gamma_{A,i} = 5/(i+1)$, $\gamma_{B,i} = 15$, and weight decay 10^{-7}.



(a) Hierachical IALM

(b) Only total-acceleration constraint

(c) Both constraints at the same level

Fig. 3. Control histograms for the expert and learned policies under the three optimization schemes. These supplementary plots help interpret how the hierarchy between pedestrian-avoidance and acceleration constraints changes the control distribution produced by the learned policy.

d) Evaluation protocol.: We compare the trilevel hierarchy, the total-acceleration-only baseline, and the flat two-constraint baseline. Evaluation is performed by closed-loop rollouts of the learned policies on randomized initial conditions close to the nominal lane. The reported results use 2000 rollouts of horizon 13 s. We report pedestrian collision rates and complement them with qualitative trajectory plots and control or acceleration histograms, shown in the main text and supplementary figures.

APPENDIX III
SUPPLEMENTARY FIGURES

Supplementary figures can be found Figure 3.