CENTRAL PATH SPO: AN INTERIOR-PROJECTION BASED APPROACH FOR SOLVER-FREE DECISION-FOCUSED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-focused learning trains predictive models for the quality of the decisions they induce, but repeated optimization during training limits its scalability. By leveraging concepts from path-following methods in convex programming and recent advances in constrained machine learning, we formulate a novel framework for solver-free decision-focused learning for linear programs. The framework is based on training a neural network to directly learn the solution of a much easier, smoothed and regularized version of the linear program via first-order methods and an interior projection map. Via Fenchel conjugacy, the learned smooth solution is transformed into a cost prediction, which is used to solve the original linear program at inference. The framework is applicable with any suitably chosen interior projection map and Legendre regularizer. We focus on the case where the interior projection map is a soft-radial projection anchored to the analytic center of the constraint set, and the regularizer is a logarithmic barrier from the classical theory of interior-point methods. The resulting method is both fast and highly performant. We derive theoretical guarantees and evaluate the method on large-scale versions of standard benchmarks, achieving lower regret and order-of-magnitude speed-ups compared with standard approaches.

1 INTRODUCTION

Optimization problems have countless real-life applications. Linear programs are especially ubiquitous, with essential applications for decision-making in areas ranging from transport system management to supply chain design and infrastructure planning (Hitchcock, 1941; Massé & Gibrat, 1957; ReVelle et al., 1969; Golden et al., 2024). Over decades, a rich literature of classical methods has developed, capable of obtaining high-precision solutions even for very large instances. An enduring challenge in applications is that an optimization model, no matter how mathematically precise, will fail to produce useful decisions if its problem parameters are ill-chosen. Under uncertain conditions, problem parameters generally have to be estimated from observed covariates, and small statistical errors may cause large deteriorations in downstream decision quality.

Decision-focused learning (DFL), also known as smart predict-and-optimize, is a recent field of machine learning that has arisen in response to this challenge. The field is highly active, with a rapidly growing literature (Mandi et al., 2024). Using modern deep learning techniques, models are trained to output predictions that directly optimize decision performance, rather than statistical accuracy.

Linear programs with random cost vectors are among the most frequently studied problem classes, owing in part to their relevance for practical applications and in part to the unique difficulty posed by their structure. Decision regret is a piecewise constant function of the prediction, making direct application of first-order methods ineffective and motivating the construction of surrogate losses. A number of successful methods have been developed, each defined by its particular choice of surrogate (Mandi et al., 2024). But a recurring bottleneck is the computational cost of training: minimizing decision regret typically requires repeated solves of optimization problems.

A substantial portion of current research focuses on this bottleneck (Zharmagambetov et al., 2023; Mulamba et al., 2021). Very recent literature aims to build solver-free approaches, using the 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.

of the underlying optimization problem to construct a decision-aligned surrogate loss that can be evaluated without solver calls (Berden et al., 2025).

A separate strand of literature in constrained machine learning addresses the same problem from a different angle. Instead of learning to map covariates to predictions intended as inputs to an optimization solver, these methods aim to map covariates directly to their associated optimal solutions (Schutte et al., 2025; Tordesillas et al., 2023; Schneider & Kuhn, 2026). Projection-based approaches replace solver calls with feasibility-preserving maps during training. These methods are highly efficient, and modern approaches can evaluate these maps without solver calls. However, while well suited to smooth problems whose optimal solutions reside in the interior of their constraint sets, they face difficulties with linear programs because their solution maps are piecewise constant and their optima generally lie on the boundary. This creates a challenge in bringing these two strands of literature together.

Our contribution. In this paper, we present a novel solver-free decision-focused learning method for linear problems. It is based on the theory of regularizers for convex programming. During training, we directly learn the optimal decision map for a highly smoothed version of the linear program. Our contributions are threefold:

- In Section 3, we introduce a general framework that converts learned interior decisions into cost predictions through Fenchel duality, with Fisher-consistency and regret guarantees.
- In Section 4, we develop CP-SPO using log-barrier regularization and soft-radial interior projection, and establish additional optimization guarantees under sufficient smoothing.
- In Section 5, we demonstrate CP-SPO's strong decision quality and computational efficiency on large-scale benchmarks, including order-of-magnitude training speed-ups over SPO+ and FY. Appendices D and E provide hyperparameter sensitivity experiments and direct-policy ablations.

2 Problem Formulation and Notation

Let $\mathcal{X} \subseteq \mathbb{R}^p$ be the covariate space and $\mathcal{C} = \mathbb{R}^d$ the cost-vector space. Covariates and costs follow a joint law $(X, C) \sim \mathbb{P}$ with compact support in $\mathcal{X} \times \mathcal{C}$. We observe a dataset $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$ of context–cost pairs. For $x \in \mathcal{X}$, the random-cost LP is

$$\min_{z \in \mathcal{Z}} \mathbb{E}[C^\top z \mid X = x], \qquad \mathcal{Z} = \{z \in \mathbb{R}^d : Az \leq b\}. \tag{1}$$

Here $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$ define a fixed, bounded feasible polytope. The goal is to learn a cost prediction map whose induced decision solves the contextual problem:

$$\hat{c} : \mathcal{X} \to \mathcal{C}, \qquad z^\star(\hat{c}(x)) \in \operatorname*{arg\,min}_{z \in \mathcal{Z}} \mathbb{E}[C^\top z \mid X = x]. \tag{2}$$

Throughout, $z^\star(c) \in \arg\min_{z \in \mathcal{Z}} c^\top z$ denotes an optimal decision. When the optimizer is not unique, it denotes the specific solution selected by the interior-point method, using the central-path limit convention detailed in Appendix A. We use full-dimensional notation for readability; affine constraints are handled by working in the affine hull (Appendix A).

3 Central-Path SPO: Solver-free decision-focused learning via interior projections

3.1 The framework

The idea is to learn an easy, smoothed decision problem during training, then recover a cost vector and solve the original LP at inference. This separates learning from the expensive optimization step. We first describe the general framework; Section 4 develops our log-barrier and soft-radial implementation.

Choose a regularizer R and a smoothing parameter $\mu > 0$. The regularized counterpart of Equation (1) is

$$\min_{z \in \mathrm{ri}(\mathcal{Z})} \mathbb{E}[C^\top z + \mu R(z) \mid X = x]. \tag{3}$$

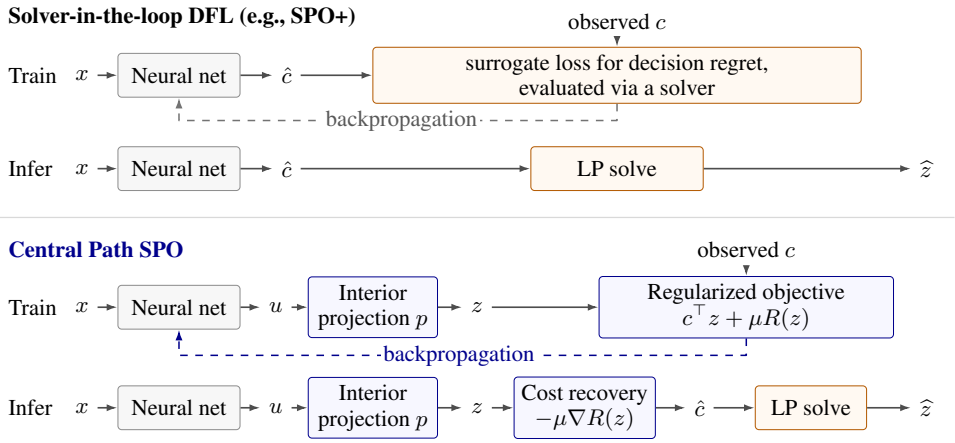

Figure 1: Solver-in-the-loop DFL evaluates a *surrogate* for regret using method-dependent optimization calls (e.g., at $2\hat{c} - c$ for SPO+). CP-SPO evaluates a projected regularized objective during training and solves the original LP at inference. Dashed arrows show backpropagation; observed c is used only in training.

We define its optimal solution map as

$$\hat{z}_\mu : \mathcal{X} \to \mathrm{ri}(\mathcal{Z}), \qquad x \longmapsto \operatorname*{arg\,min}_{z \in \mathrm{ri}(\mathcal{Z})} \mathbb{E}[C^\top z + \mu R(z) \mid X = x]. \tag{4}$$

For a fixed cost, write

$$\ell_\mu^R(z, c) = c^\top z + \mu R(z), \qquad z_\mu^R(c) = \operatorname*{arg\,min}_{z \in \mathrm{ri}(\mathcal{Z})} \ell_\mu^R(z, c). \tag{5}$$

We use smooth Legendre regularizers with positive-definite Hessian, so these minimizers are unique and interior. The precise assumptions are in Appendix A.

Training. A neural network produces a raw output $u = g_\theta(x)$. An interior projection map p turns it into a feasible decision $\pi_\theta(x) = p(g_\theta(x))$. We train this policy directly on the regularized objective:

$$\min_\theta \ \widehat{\mathcal{L}}_\mu(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[c_i^\top \pi_\theta(x_i) + \mu R(\pi_\theta(x_i)) \right]. \tag{6}$$

When the interior projection and regularizer admit inexpensive evaluations and derivatives, this uses ordinary backpropagation without a solver in the training loop.

Inference. The smoothed decision is not our final LP decision. Its first-order condition, or equivalently Fenchel duality, gives

$$z_\mu^R(c) = \nabla R^\star(-c/\mu), \qquad c = -\mu \nabla R(z_\mu^R(c)), \tag{7}$$

where $R^\star(v) = \sup_{z \in \mathrm{ri}(\mathcal{Z})} \{v^\top z - R(z)\}$. Thus a gradient evaluation converts the learned interior decision into a cost prediction, which is passed to the original LP:

$$\hat{c}_\theta(x) = -\mu \nabla R(\pi_\theta(x)), \tag{8}$$

$$\widehat{z}_\theta(x) = z^\star(\hat{c}_\theta(x)) \in \operatorname*{arg\,min}_{z \in \mathcal{Z}} \hat{c}_\theta(x)^\top z. \tag{9}$$

We can therefore learn a strongly smoothed interior solution without deploying that smoothed solution. The parameter μ stays fixed during training; it need not approach zero. In practice, we select it using validation regret. Its positive factor in Equation (8) can be omitted without changing the LP decision.

Comparison with solver-in-the-loop DFL. Conventional methods predict costs directly and train using a surrogate for the regret induced by those costs. Schematically, for an observed cost c and a prediction c',

$$L(c, c') \;\approx\; \underbrace{c^\top z^\star(c') - \min_{z \in \mathcal{Z}} c^\top z}_{\text{regret}}. \tag{10}$$

Here $\approx$ denotes using a surrogate, not an approximation guarantee. The surrogate determines the solver calls and the gradients used for training. SPO+, for example, upper-bounds regret and obtains a subgradient from an optimizer at $2c' - c$, together with an optimizer at c (Elmachtoub & Grigas, 2022). Figure 1 contrasts this with our solver-free training objective.

The framework is not tied to one regularizer or interior projection map. Compatible alternatives from optimization learning can be used, provided the interior projection map can represent the regularized solutions. Cost recovery requires the regularizer conditions above; an optimization guarantee additionally requires control of the interior projection's Jacobian, not feasibility alone. We choose a logarithmic barrier and a geometrically centered soft-radial interior projection because their geometry makes these conditions explicit.

Heuristic extension. The framework can also be extended heuristically to mixed-integer linear programs by training on their LP relaxations and solving the original problem under the recovered cost at deployment. We use this extension for the knapsack and facility-location benchmarks. All theoretical results in this paper concern LPs only.

3.2 Theoretical guarantees

We analyze the random-cost LP of Section 2. Let $z_*(x) = \hat{z}_\mu(x) = z_\mu^R(\mathbb{E}[C \mid X = x])$. Define the population excess training loss and the regret of the recovered LP decisions by

$$\mathcal{E}_\mu = \mathbb{E}[\ell_\mu^R(\pi_\theta(X), C) - \ell_\mu^R(z_*(X), C)], \tag{11}$$

$$\mathcal{E} = \mathbb{E}\left[\mathbb{E}[C \mid X]^\top \widehat{z}_\theta(X) - \min_{z \in \mathcal{Z}} \mathbb{E}[C \mid X]^\top z\right]. \tag{12}$$

Theorem 1 (Regret from excess training loss)**.** *Fix $\mu > 0$, suppose $\|\mathbb{E}[C \mid X = x]\| \le B_C$ almost surely, with $B_C > 0$, and let $D = \operatorname{diam}_2(\mathcal{Z})$. Set*

$$\Lambda_\mu = \sup_{\|a\| \le 2B_C} \|\nabla^2 R(z_\mu^R(a))\|_{\mathrm{op}} < \infty. \tag{13}$$

Every measurable interior policy with finite excess training loss satisfies

$$\mathcal{E} \le D\sqrt{2\mu\Lambda_\mu \mathcal{E}_\mu}. \tag{14}$$

The constant is finite because bounded costs map to a compact interior region. No bound on the raw outputs or recovered costs is needed, and there is no additive smoothing bias. The proof uses an exact Bregman expression for the excess loss (Appendix B.2).

Corollary 2 (Fisher consistency)**.** *At any fixed $\mu > 0$, exact conditional minimization recovers $\hat{c}_\theta(x) = \mathbb{E}[C \mid X = x]$ and hence an optimal decision for the conditional expected objective. In particular, $\mathcal{E}_\mu = 0$ implies $\mathcal{E} = 0$.*

Cost recovery follows directly from the first-order condition; the affine-constraint version is given in Appendix B.1.

Optimization guarantees. As we shall see, output-space optimization guarantees can also be obtained for the framework under suitable conditions on the regularizer and interior projection map. We prove such a guarantee for our specific choices in Section 4.3. See Appendix B.3 for details.

4 Our Specific Method

We now choose the regularizer and interior projection map. The log barrier makes linear programs smooth in a way that is adapted to their constraints. Soft-radial projection lets us learn those smoothed solutions through an explicit interior projection map.

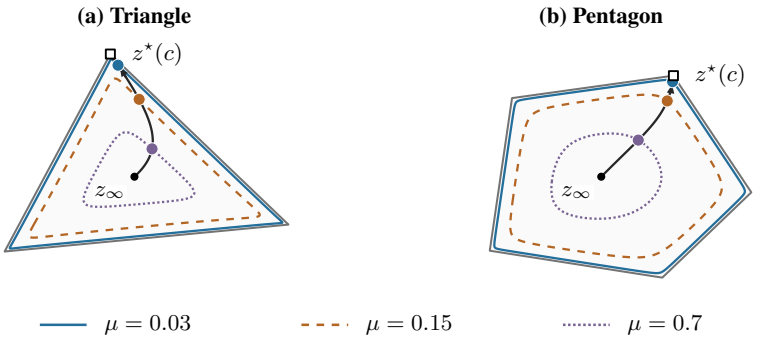

Figure 2: Log-barrier smoothing on two polytopes. Each colored curve contains the smoothed solutions $z_\mu(c)$ as the *unit-norm* cost vector varies, at one of three smoothing levels. Stronger smoothing concentrates these solutions near the analytic center z_∞. The black curve traces the central path for one fixed cost; its marked points correspond to the same three values of μ, and the square marks the limiting LP optimum.

4.1 Our Chosen Regularizer: A Logarithmic Barrier

Interior-point methods are a central approach to solving linear programs (Boyd & Vandenberghe, 2004). Their basic device is the logarithmic barrier

$$V(z) = -\sum_{j=1}^{m} \log(b_j - a_j^\top z), \qquad Az < b, \tag{15}$$

where $a_j^\top$ is row j of A. Each term grows without bound as its constraint becomes tight. Adding μV therefore repels solutions from the boundary, replacing the LP's piecewise-constant solution map with a smooth one:

$$z_\mu(c) = \arg\min_{Az<b} \{c^\top z + \mu V(z)\}. \tag{16}$$

Here $z_\mu = z_\mu^V$ in the framework's notation. Large μ emphasizes the barrier, pulling solutions towards the interior; small μ emphasizes the linear objective, moving them towards the LP optimum.

For a fixed cost c, varying μ traces its *central path*. As $\mu \downarrow 0$, the path approaches an LP optimum. At the other end, it approaches the *analytic center*, the minimizer of the barrier:

$$z_\infty = \arg\min_{Az<b} V(z), \qquad \lim_{\mu\to\infty} z_\mu(c) = z_\infty. \tag{17}$$

Figure 2 shows both effects: changing the cost moves the smoothed solution around the interior, while changing μ moves it along its central path. Classical path-following methods progressively decrease μ, using Newton steps to follow this path to an LP solution. CP-SPO instead learns at a fixed smoothing level and recovers the cost.

Recovering the cost. For the log barrier, the inverse map in Equation (7) has the explicit form

$$c_\mu(z) = -\mu \nabla V(z) = -\mu \sum_{j=1}^{m} \frac{a_j}{b_j - a_j^\top z}. \tag{18}$$

It only requires evaluating constraint slacks and a weighted sum of their normals. We do not need to learn a point close to the LP optimum: a point well inside the feasible set still identifies its cost through this map. Under our interior-point selection convention, solving the LP under the recovered cost returns the optimal decision at the end of the same central path, even when the LP has multiple optima.

4.2 Our Chosen Interior Projection: Soft Radial Projection

Hard radial projection is a simple projection method for recovering a feasible point in a convex constraint set. It projects along rays from an interior reference point, rather than minimizing Euclidean distance.

Soft-radial projection (Schneider & Kuhn, 2026) modifies this construction to return strictly interior points. It was designed to avoid the zero-gradient pathologies of the hard radial map: all exterior points on a ray are mapped to the same boundary point, eliminating variation in that direction. The soft variant preserves this variation while ensuring strict feasibility, making it an excellent choice for Central Path SPO and our log-barrier objective.

Formally, let $\mathcal{Z} = \{z : Az \leq b\}$ and fix an interior reference point u_0. The hard radial map is

$$q(u) = u_0 + \frac{u - u_0}{\max\left\{1, \max_{1 \leq j \leq m} \frac{a_j^\top (u - u_0)}{b_j - a_j^\top u_0}\right\}}. \tag{19}$$

The soft-radial interior projection map further contracts towards u_0:

$$p(u) = u_0 + r\big(\|u - u_0\|^2\big)\,\big(q(u) - u_0\big). \tag{20}$$

Here r is a fixed, smooth, strictly increasing function with values in $(0, 1)$; for example, $r(s) = \varepsilon + (1 - \varepsilon)s/(s + \tau)$, with fixed parameters $0 < \varepsilon < 1$ and $\tau > 0$. The contraction varies with distance from u_0, keeping $p(u)$ strictly interior without collapsing exterior rays (Figure 3). The interior projection map is differentiable almost everywhere.

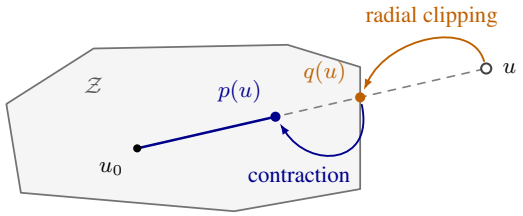

Figure 3: Soft-radial interior projection. Radial clipping sends an infeasible point u to $q(u)$ on the boundary; contraction towards the interior reference point u_0 gives $p(u)$.

Geometric centering. We choose $u_0 = z_\infty$: the analytic center minimizes our barrier and is where the smoothed solutions concentrate as μ increases. We compute it once and scale the coordinates using the barrier Hessian there. In these centered and scaled coordinates, the center is 0 and

$$V(y) = -\sum_{j=1}^{m} \log(1 - a_j^\top y), \qquad \nabla V(0) = 0, \qquad \nabla^2 V(0) = I. \tag{21}$$

Here we reuse a_j for the transformed constraint normals; an additive barrier constant is omitted. This makes the unit ball feasible and sets a natural scale for raw outputs. On any ball of radius $\rho < 1$, hard radial clipping is inactive, so the interior projection is simply $p(u) = r(\|u\|^2)u$. The next result gives a sufficient smoothing level at which restricting outputs to such a ball preserves the optimum. Implementation details are in Appendix C.

4.3 Guarantees for the Chosen Method

Regret. With $R = V$, Theorem 1 gives the explicit bound

$$\mathcal{E} \leq D\left(m + \frac{2B_C r_0}{\mu}\right) \|A_0\|_{\mathrm{op}} \sqrt{2\mu\mathcal{E}_\mu}. \tag{22}$$

where $r_0 = \max_{z \in \mathcal{Z}} \|z - z_\infty\|$ and row j of A_0 is $a_j^\top / (b_j - a_j^\top z_\infty)$ in the original affine-hull coordinates. The derivation is in Appendix B.2; Fisher consistency follows from the same general argument.

Optimization under sufficient smoothing. Generally, soft-radial interior projection yields an invex loss in the unbounded coordinate, when solving a smooth, constrained, strictly convex optimization problem. This means that the gradient is zero only at the global optimum. However, a global Polyak–Łojasiewicz (PL) inequality generally does not hold, meaning that gradients so small that

they are numerically zero is still a possibility (Schneider & Kuhn, 2026, Section 3). For our log-barrier smoothed problem, this problem is avoided, and we obtain a PL-guarantee. This highlights an particularly favorable synergy between our interior projection map and choice of regularizer. Our barrier objective places the optimum in a region where the interior projection has uniformly controlled derivatives. Work in the normalized coordinates of Equation (21), with the reference point at 0. Fix $0 < \varepsilon < 1$ and a smooth increasing function $r_\varepsilon : [0, \infty) \to [\varepsilon, 1)$ satisfying $r_\varepsilon(0) = \varepsilon$ and $\lim_{s\to\infty} r_\varepsilon(s) = 1$. Define

$$\begin{aligned} F_{\mu,c}(y) &= c^\top y + \mu V(y), \\ p_\varepsilon(u) &= r_\varepsilon(\|u\|^2) q(u), \qquad f_{\mu,c,\varepsilon} = F_{\mu,c} \circ p_\varepsilon. \end{aligned} \tag{23}$$

Here $p_\varepsilon : \mathbb{R}^d \to \mathrm{ri}(\mathcal{Z})$, and q is the hard radial map with anchor 0 in these coordinates. We write $p = p_\varepsilon$ and $r = r_\varepsilon$ when the contraction choice is fixed. The cost c is transformed too: $c = H_0^{-1/2} c_{\mathrm{orig}}$, where H_0 is the barrier Hessian at the analytic center in the original affine-hull coordinates.

Theorem 3 (PL inequality)**.** *Fix $\rho \in (0, 1)$, set $\rho_p = \rho r_\varepsilon(\rho^2)$, and suppose $\|c\| \le C_{\max}$. If $\mu > 0$ satisfies*

$$\mu \ge C_{\max} \left(1 + \frac{1}{\rho r_\varepsilon(\rho^2)} \right), \tag{24}$$

then an unrestricted minimizer $u^\star_{\mu,c,\varepsilon}$ lies in $\{u : \|u\| \le \rho\}$. Restricting raw outputs to this ball therefore preserves the unrestricted optimum $f^\star_{\mu,c,\varepsilon} = \min_{u\in\mathbb{R}^d} f_{\mu,c,\varepsilon}(u)$. Throughout this ball,

$$\tfrac{1}{2} \|\nabla_u f_{\mu,c,\varepsilon}(u)\|^2 \ge \frac{\mu\varepsilon^2}{(1+\rho_p)^2} \big(f_{\mu,c,\varepsilon}(u) - f^\star_{\mu,c,\varepsilon} \big). \tag{25}$$

Moreover, its optimal output obeys

$$\|u^\star_{\mu,c,\varepsilon}\| \le \frac{C_{\max}}{\varepsilon(\mu - C_{\max})}. \tag{26}$$

Stronger smoothing relative to cost magnitude brings the required outputs closer to the center. In that region, Lemma B.1 applies with $\alpha = \mu/(1+\rho_p)^2$ and $\sigma = \varepsilon$. This remains an output-space guarantee; transferring it to neural-network weights requires an additional network-Jacobian condition. The proof is in Appendix B.3.

5 Empirical Evaluation

We evaluate on shortest path, knapsack, and facility location, three canonical problem classes in the SPO literature, using the benchmark implementations of Mandi et al. (2025). We select instances from their evaluation of larger problems and add the unstructured, randomly generated LPs introduced by LAVA (Berden et al., 2025). Our benchmarks include large-scale instances by the standard of the DFL literature: for context, Eilles-Chan Way et al. (2026) describe 300-variable knapsack and 400-variable quadratic portfolio single-scenario instances as approximately six and eight times larger, respectively, than typical prior benchmarks. Our largest knapsack instance has 500 variables, and our largest facility location instance has 2500. Our aim is to examine whether solver-free training allows DFL to scale more effectively. We use a modest 100 training context–cost pairs, since data-scarce regimes are a principal motivation for DFL, with 32 further pairs for validation.

LAVA trains quickly, but requires an expensive preprocessing step. For some of our instances, this was not computationally feasible, and a one-hour time limit was reached.

Table 1: Normalized expected test regret (%): mean and sample SD (second line) over ten matched runs, each with 100 training pairs, 32 validation pairs, and 128 test contexts. Bold: lowest mean; dash: unavailable LAVA geometry.

Size	CP-SPO	MSE	SPO+	WISE	FY	NCE	LAVA
Shortest path							
	8.25	13.43	19.44	10.46	19.14	41.78	29.08
10×10	±1.19	±1.80	±3.58	±1.09	±2.68	±9.09	±5.44
	8.04	15.13	20.18	14.64	24.86	45.68	37.70
15×15	±1.40	±1.37	±2.95	±2.16	±2.74	±6.64	±4.92
	7.97	13.21	20.11	14.14	24.80	47.35	
20×20	±0.71	±1.52	±1.29	±1.46	±2.28	±2.56	—
	7.90	17.02	25.03	15.51	32.02	50.91	
25×25	±0.74	±1.65	±2.42	±1.34	±2.07	±5.71	—
Facility Location							
	10.76	12.13	**9.41**	21.82	11.04	107.31	15.67
50×5	±1.04	±0.83	±0.53	±0.95	±0.65	±21.05	±1.20
	16.61	17.05	18.25	37.65	23.28	179.63	
100×10	±0.60	±1.18	±1.07	±2.30	±0.98	±43.69	—
	18.96	**18.94**	20.06	44.61	23.45	200.70	
200×10	±1.27	±0.89	±0.81	±2.52	±1.04	±38.80	—
	20.58	**20.44**	22.92	50.53	25.86	222.25	
250×10	±1.45	±0.86	±1.47	±2.58	±1.57	±48.67	—
Random LP							
	2.01	3.82	3.03	**0.54**	4.31	18.82	4.10
50×50	±0.60	±0.92	±0.93	±0.32	±1.26	±3.11	±1.27
	1.43	3.17	3.57	**0.71**	4.62	24.22	5.60
150×50	±0.47	±1.14	±1.54	±0.35	±1.91	±6.40	±1.91
	1.51	3.34	3.93	**0.68**	4.83	25.74	6.02
200×50	±0.39	±0.85	±1.23	±0.17	±1.34	±3.52	±1.52
	1.22	4.17	5.58	**0.69**	6.51	26.75	8.30
400×50	±0.32	±1.42	±1.62	±0.43	±1.35	±7.02	±1.51
Knapsack							
	9.92	13.28	11.31	11.42	15.76	38.96	15.75
50×2	±1.43	±2.49	±1.06	±1.44	±2.23	±4.26	±2.21
	9.15	11.40	10.96	11.23	17.41	34.67	15.98
100×2	±1.15	±0.87	±1.03	±1.15	±1.47	±3.07	±1.67
	9.53	11.72	12.45	12.56	15.43	36.26	17.34
200×2	±0.48	±0.91	±0.49	±0.94	±1.03	±3.54	±1.02
	9.18	11.95	11.16	12.15	14.44	36.39	16.41
500×2	±0.71	±0.90	±0.71	±0.74	±0.84	±3.28	±0.80

We compare CENTRAL PATH SPO (CP-SPO) with six baselines: MSE, which minimizes cost-prediction error; SPO+, a convex regret surrogate (Elmachtoub & Grigas, 2022); FY, a perturbed Fenchel–Young loss (Berthet et al., 2020); NCE, which contrasts observed optimal decisions with cached alternatives (Mulamba et al., 2021); WISE, a solver-free weighted direction-regression loss (Wan & Liu, 2026); and LAVA, which learns from adjacent vertices (Berden et al., 2025). We use PyEPO (Tang & Khalil, 2022) with HiGHS for SPO+, FY, and NCE, reuse LAVA's published implementation, and implement MSE and WISE ourselves. Methods share data, seed bundles, and the joint architecture/optimizer options within each instance; loss-specific parameter grids differ. Each is selected by validation regret on the corresponding benchmark problem. Table 1 reports regret on 128 test contexts over ten matched runs; Appendix C gives the protocol.

LAVA requires preprocessing to compute vertices adjacent to the training-set optimal solutions. This stage reached our one-hour limit on the larger shortest-path and facility-location instances, leaving the corresponding results unavailable. Expensive neighbor enumeration at degenerate vertices is a known limitation of the method (Berden et al., 2025).

Table 2: Mean training wall time (seconds) over all available settings and the 11 common settings. Optimizer-loop time only; preparation is separate. Full measurements appear in Appendix C.

Method	CP-SPO	MSE	SPO+	WISE	FY	NCE	LAVA
Mean (s)	3.158	0.988	70.479	0.462	49.634	0.326	123.390
Common 11 (s)	1.363	0.905	50.241	0.649	29.915	0.295	123.390

Performance across the selected benchmarks. Across our benchmarks and against the selected baselines, CP-SPO achieves strong and consistent performance, though it is not uniformly best. It has the lowest mean regret in 9 of the 16 settings: all four shortest-path grids, all four knapsack instances, and one facility-location instance. Notably, our method achieves highly consistent performance. It is the only DFL method with lower average regret than basic MSE in all four problem families.

WISE clearly outperforms CP-SPO on all four random LPs, but CP-SPO beats every other baseline on each of these instances. At 400×50, WISE achieves 0.69% regret, compared with 1.22% for CP-SPO and 5.58% for SPO+. Conversely, WISE struggles on facility location: at 250×10, its regret is 50.53%, compared with 20.58% for CP-SPO. CP-SPO remains competitive there, with MSE narrowly ahead at 20.44%. The direct-policy ablation in Appendix E examines the role of learning a smoothed interior policy and recovering a cost, rather than directly deploying the learned policy.

Training time. Most importantly, avoiding solver-in-the-loop training gives CP-SPO order-of-magnitude speed-ups over the standard SPO+ and FY baselines. Across the 16 settings, mean training time is 3.16 seconds for CP-SPO, versus 70.48 for SPO+ and 49.63 for FY: ratios of approximately $22\times$ and $16\times$, respectively. We remark that pure regression methods, such as WISE and MSE, remain faster. NCE is fast, but this is mostly due to early epoch selection, as the method struggles to train effectively on these large-scale, small-data instances. Tables 2 and 3 measure optimizer-loop wall time for each method's selected configuration and training horizon.

6 Conclusion

We propose a solver-free framework for decision-focused learning that enables the use of techniques from optimization learning and constrained machine learning.

Our theoretical analysis assumes a linear program with no integer variables. Although our experiments also show strong performance on problems with integer variables, future work could specialize the method for such instances and derive corresponding theoretical bounds.

Our method uses log-barrier smoothing for regularization and a soft radial projection as the interior projection map. We establish a strong synergy between these choices, while alternative combinations remain a direction for future work. We expect the framework to be applicable beyond the problems considered in this paper and the specific method developed here.

Reproducibility Statement

Appendix C documents the experimental protocol, including data splits, hyperparameter selection, evaluation metrics, and timing procedures. Tables 7–13 report the selected hyperparameter configurations for each method and instance, and Algorithm 1 specifies our method. The mathematical assumptions and proofs are provided in Appendices A and B. We use existing implementations from PyEPO (Tang & Khalil, 2022) for SPO+, FY, and NCE, and the published LAVA implementation (Berden et al., 2025); our own implementations are limited to CP-SPO and the straightforward MSE and WISE baselines. An associated software package for CP-SPO will be released after publication. The code for reproducing the experiments is available upon request.

AI Use Statement

AI tools were used for assistance in the development of ths paper, though they did not make any essential contributions.

Our framework and specific method, along with their associated theory, were developed without the assistance of AI tools.

AI did not play a vital role in the writing of the final paper either.

Implementation:

- As is standard in modern software development, AI tools assisted in the writing of the code, with human supervision and review. The code structure, algorithms and experments were designed by humans without AI assistance.
- The projecti involved running a large number of numerically heavy experiments on remote clusters. Frontier AI tools were used in the orchestration and management of remote CPUs. This was its most significant contribution to the project.

Writing:

- AI tools were not involved in the design of the paper's outline, or the writing of its first version. AI was used for text editing, constructinglatex expressions, formatting, and critical review.

Literature:

- Frotier AI tools were used to stress test the contribution against existing literatur, and to find relevant references. No references found by the AI were previously uknown to the authors, however, and so AI tools did not impact the appearance of the bibliography in the end.

Theory:

- AI did not perticipate in the writing of the proofs, nor in the design of the algorithms.
- AI was used to assess the final results for correctness. No errors were detected by AI at any point, other than notational issues.

REFERENCES

Senne Berden, Ali İrfan Mahmutoğulları, Dimos Tsouros, and Tias Guns. Solver-free decision-focused learning for linear optimization problems. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5284.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2002.08676.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with fenchel–young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020. URL https://jmlr.org/papers/v21/19-021.html.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Stéphane Eilles-Chan Way, Hugo Percot, Quentin Cappart, Tias Guns, and Louis-Martin Rousseau. Scaling decision-focused learning to large problems with lagrangian decomposition. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2217–2226, 2026. doi: 10.24963/ijcai.2026/247. URL https://www.ijcai.org/proceedings/2026/247.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Bruce Golden, Linus Schrage, Douglas Shier, and Lida Anna Apergi. The unexpected power of linear programming: An updated collection of surprising applications. *Annals of Operations Research*, 343:573–605, 2024. doi: 10.1007/s10479-024-06245-5.

Frank L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20(1–4):224–230, 1941. doi: 10.1002/sapm1941201224.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Víctor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. doi: 10.1613/JAIR.1.15320.

Jayanta Mandi, Ali İrfan Mahmutoğulları, Senne Berden, and Tias Guns. Minimizing surrogate losses for decision-focused learning using differentiable optimization. In *ECAI 2025*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 3888–3895, 2025.

P. Massé and R. Gibrat. Application of linear programming to investments in the electric power industry. *Management Science*, 3(2):149–166, 1957. doi: 10.1287/mnsc.3.2.149.

Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Víctor Bucarey, and Tias Guns. Contrastive losses and solution caching for predict-and-optimize. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 2833–2840, 2021. doi: 10.24963/ijcai.2021/390.

Charles ReVelle, Erhard Joeres, and William H. Kirby. The linear decision rule in reservoir management and design: 1, development of the stochastic model. *Water Resources Research*, 5(4): 767–777, 1969. doi: 10.1029/WR005i004p00767.

Philipp J. Schneider and Daniel Kuhn. Soft-radial projection for constrained end-to-end learning. *arXiv preprint arXiv:2602.03461*, 2026. URL https://arxiv.org/abs/2602.03461.

Noah Schutte, Grigorii Veviurko, Krzysztof Postek, and Neil Yorke-Smith. Sufficient decision proxies for decision-focused learning. *arXiv preprint arXiv:2505.03953*, 2025. URL https://arxiv.org/abs/2505.03953.

Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *arXiv preprint arXiv:2206.14234*, 2022. URL https://arxiv.org/abs/2206.14234.

Jesus Tordesillas, Jonathan P. How, and Marco Hutter. RAYEN: Imposition of hard convex constraints on neural networks. *arXiv preprint arXiv:2307.08336*, 2023. URL https://arxiv.org/abs/2307.08336.

Beichen Wan and Mo Liu. A solver-free training method for predict-then-optimize. *arXiv preprint arXiv:2606.19587*, 2026. URL https://arxiv.org/abs/2606.19587.

Arman Zharmagambetov, Brandon Amos, Aaron Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. Landscape surrogate: Learning decision losses for mathematical optimization under partial information. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1191.

A MATHEMATICAL SETTING

We work with a fixed, compact polytope

$$\mathcal{Z} = \{z \in \mathbb{R}^d : Az \le b,\ Ez = e\}.$$

The equalities describe its affine hull, and there is a point satisfying $Az < b$ and $Ez = e$. Constraints that hold as equalities throughout $\mathcal{Z}$ are included in $Ez = e$, not in the barrier. These conventions permit the log barrier to be defined on the relative interior:

$$V(z) = -\sum_{j=1}^{m} \log(b_j - a_j^\top z), \qquad z \in \mathrm{ri}(\mathcal{Z}).$$

The main text uses full-dimensional notation to keep the presentation simple.

Let N have orthonormal columns spanning $\ker E$, and choose any z_{ref} with $Ez_{\mathrm{ref}} = e$. In reduced coordinates $z = z_{\mathrm{ref}} + Ny$, write

$$\widetilde{V}(y) = V(z_{\mathrm{ref}} + Ny), \qquad \widetilde{c} = N^\top c.$$

The reduced feasible set is full-dimensional and bounded. Its barrier Hessian is

$$\nabla^2 \widetilde{V}(y) = N^\top A^\top \operatorname{diag}\big((b_j - a_j^\top z)^{-2}\big)\, AN \succ 0.$$

Indeed, a nonzero null direction would leave every inequality unchanged and contradict boundedness. The smoothed problem therefore has a unique minimizer for every cost and every $\mu > 0$.

General regularizers. For the consistency and regret results, work in these orthonormal affine-hull coordinates and extend R to a proper, lower-semicontinuous convex function on the ambient reduced space. Its domain has interior $\operatorname{int} \mathcal{Z}$ and closure $\mathcal{Z}$. Assume that R is Legendre: it is strictly convex and differentiable on the interior, and its gradient norm diverges when an interior sequence approaches the boundary. For the regret result, also assume that R is C^2 with positive-definite Hessian on the interior. These conditions hold for the log barrier above. If the affine hull is zero-dimensional, every feasible decision is identical and the regret statement is immediate; otherwise we use positive dimension.

Boundedness of $\mathcal{Z}$ and an affine lower bound on R make R^* finite everywhere. Legendre duality gives the unique interior solution and the cost-recovery identity:

$$z_\mu^R(c) = \nabla R^*(-c/\mu), \qquad -\mu \nabla R(z_\mu^R(c)) = c.$$

The solution map is continuous; under the C^2 positive-Hessian assumption, it is continuously differentiable by the inverse function theorem.

Optimal-solution selection. For the unregularized LP, $z^\star$ denotes a fixed measurable selection from the optimal set. Our interior-point convention is the limiting log-barrier solution

$$z^\star(c) = \lim_{\tau \downarrow 0} z_\tau(c) \in \arg\min_{z \in \mathcal{Z}} c^\top z,$$

with the barrier restricted to the affine hull as above. This limit exists for the bounded polytope and may lie on an optimal face rather than at a vertex. Positive cost rescaling preserves the selection: $z_\tau(\alpha c) = z_{\tau/\alpha}(c)$ for $\alpha > 0$, hence $z^\star(\alpha c) = z^\star(c)$. At $c = 0$, the selection is z_∞. The consistency and regret results below hold for any fixed measurable optimal selection, not only this interior-point convention; they require neither a unique LP optimizer nor a particular numerical solver.

We assume $\mathbb{E}\|C\| < \infty$, so that $\mathbb{E}[C \mid X]$ exists almost surely. Policies and their induced decision maps are measurable, with well-defined finite risk. Fisher consistency concerns attaining the unrestricted conditional minimizer; the regret bound applies to any interior policy with finite excess risk. Training and deployment use the same feasible polytope $\mathcal{Z}$.

B Theoretical Results and Proofs

B.1 Fisher Consistency

Proof of Corollary 2. Conditioning the population loss on $X = x$ replaces C by $\mathbb{E}[C \mid X = x]$. The unique conditional minimizer is $z_*(x) = z_\mu^R(\mathbb{E}[C \mid X = x])$. Its first-order condition in the affine-hull coordinates is

$$\mathbb{E}[C \mid X = x] + \mu \nabla R(z_*(x)) = 0.$$

Thus the recovered cost is $\mathbb{E}[C \mid X = x]$, and optimizing it over $\mathcal{Z}$ is exactly optimizing the conditional expected objective. Every selected optimizer is therefore conditionally optimal, including when there are ties. If a policy class attains the unrestricted population minimum, nonnegativity and strict convexity of the conditional excess imply that its policy equals z_* almost surely.

In the original coordinates, equality of the reduced costs means that their difference lies in $\mathrm{range}(E^\top)$. This changes the objective by a constant on $Ez = e$, so the optimal sets coincide and the selected decision remains optimal. The argument holds for each fixed $\mu > 0$ and does not require the training parameter to tend to zero. □

B.2 Regret from Excess Training Loss

We use the general regularizer and affine-hull coordinates of Appendix A. Fix $\mu > 0$ and a measurable interior policy $z(x) = \pi_\theta(x)$. Write

$$\begin{aligned} c_*(x) &= \mathbb{E}[C \mid X = x], \qquad & z_*(x) &= z_\mu^R(c_*(x)), \\ \hat{c}(x) &= -\mu \nabla R(z(x)), \qquad & y(x) &= z^\star(\hat{c}(x)) \in \operatorname*{arg\,min}_{w \in \mathcal{Z}} \hat{c}(x)^\top w. \end{aligned}$$

The conditional excess training loss is

$$\Delta_\mu(x) = c_*(x)^\top (z(x) - z_*(x)) + \mu\big(R(z(x)) - R(z_*(x))\big), \qquad \mathcal{E}_\mu = \mathbb{E}[\Delta_\mu(X)]. \tag{27}$$

The comparison is with the unrestricted conditional minimum. No assumption that the learned policy belongs to a class containing z_* is needed.

B.2.1 The Exact Excess-Loss Identity

For a differentiable convex function G, let

$$D_G(v, w) = G(v) - G(w) - \nabla G(w)^\top (v - w), \qquad H_\mu(c) = \mu R^*(-c/\mu).$$

At a fixed context, abbreviate $c = c_*(x)$, $\hat{c} = \hat{c}(x)$, $z = z(x)$, and $z_* = z_*(x)$. The first-order condition $c + \mu \nabla R(z_*) = 0$ gives

$$\Delta_\mu(x) = \mu D_R(z, z_*) = D_{H_\mu}(c, \hat{c}). \tag{28}$$

For the second equality, use $z = z_\mu^R(\hat{c})$, $\nabla H_\mu(\hat{c}) = -z$, and $H_\mu(\hat{c}) = -\hat{c}^\top z - \mu R(z)$, and expand the divergence. This is the Fenchel–Young/Bregman correspondence (Blondel et al., 2020). The label-only term $H_\mu(C)$ need not be evaluated during training. Differentiating the inverse-gradient relation also gives

$$\nabla^2 H_\mu(a) = \frac{1}{\mu} [\nabla^2 R(z_\mu^R(a))]^{-1}. \tag{29}$$

B.2.2 Proof of Theorem 1

The map z_μ^R is continuous and maps every finite cost to an interior point. Thus the image of the compact cost ball $\{a : \|a\| \le 2B_C\}$ is a compact subset of the interior. Continuity and positive definiteness of $\nabla^2 R$ imply $0 < \Lambda_\mu < \infty$.

Fix a context with $\|c\| \le B_C$ and choose $y^* \in \arg\min_{w \in \mathcal{Z}} c^\top w$. Optimality of y under $\hat{c}$ gives $\hat{c}^\top (y - y^*) \le 0$. With $D = \mathrm{diam}_2(\mathcal{Z})$, both

$$c^\top (y - y^*) \le (c - \hat{c})^\top (y - y^*) \le D\|c - \hat{c}\|, \qquad c^\top (y - y^*) \le D\|c\| \le DB_C$$

hold. Consequently

$$\mathrm{Reg}_c(y) := c^\top y - \min_{w \in \mathcal{Z}} c^\top w \le D \min\{\|c - \hat{c}\|, B_C\}. \tag{30}$$

If $\hat{c} = c$, both the regret and the excess are zero. Otherwise set $d_c = \|\hat{c} - c\|$, $v = (\hat{c} - c)/d_c$, and $s = \min\{d_c, B_C\}$. Taylor's integral identity, with its origin at c, reads

$$D_{H_\mu}(c, \hat{c}) = \int_0^{d_c} t\, v^\top \nabla^2 H_\mu(c + tv) v \, dt.$$

For $0 \le t \le s$, one has $\|c + tv\| \le 2B_C$. Hence Equation (29) and the definition of Λ_μ give $\nabla^2 H_\mu(c + tv) \succeq (\mu\Lambda_\mu)^{-1} I$ on this part of the segment. The rest of the integral is nonnegative by convexity, so

$$\Delta_\mu(x) = D_{H_\mu}(c, \hat{c}) \ge \frac{s^2}{2\mu\Lambda_\mu} = \frac{\min\{\|c - \hat{c}\|, B_C\}^2}{2\mu\Lambda_\mu}. \tag{31}$$

Combining this with Equation (30) proves

$$\mathrm{Reg}_{c_*(x)}(y(x)) \le D\sqrt{2\mu\Lambda_\mu \Delta_\mu(x)}.$$

Taking expectations and applying Jensen's inequality proves Equation (14). No bound on $\hat{c}$ or on the raw network output was used. If $c_* = 0$ almost surely, the decision regret is zero directly. The proof uses only optimality of the selected decision y under $\hat{c}$, not uniqueness, and therefore holds for every optimal tie-breaking rule satisfying the measurability convention.

B.2.3 An Explicit Constant for the Log Barrier

Let $R = V$, $s_j(z) = b_j - a_j^\top z$, and let z_∞ be the analytic center in orthonormal affine-hull coordinates. Set

$$r_0 = \max_{z \in \mathcal{Z}} \|z - z_\infty\|, \qquad (A_0)_{j,:} = \frac{a_j^\top}{s_j(z_\infty)}.$$

For $z = z_\mu(a)$ with $\|a\| \le Q$, stationarity implies

$$\sum_{j=1}^m \frac{s_j(z_\infty)}{s_j(z)} = m + \nabla V(z)^\top (z - z_\infty) = m - \frac{a^\top (z - z_\infty)}{\mu} \le m + \frac{Q r_0}{\mu}.$$

Every summand is positive, so each ratio is at most this upper bound. It follows that

$$\nabla^2 V(z) = A_0^\top \operatorname{diag}\left(\left[\frac{s_j(z_\infty)}{s_j(z)}\right]^2\right) A_0 \preceq \left(m + \frac{Q r_0}{\mu}\right)^2 \|A_0\|_{\mathrm{op}}^2 I.$$

Taking $Q = 2B_C$ gives

$$\Lambda_\mu \le \left(m + \frac{2B_C r_0}{\mu}\right)^2 \|A_0\|_{\mathrm{op}}^2. \tag{32}$$

Substituting this into Equation (14) proves Equation (22). The learning and deployment argument is unchanged.

Observed-cost regret. The theorem concerns regret relative to the best decision given the context. If $v_{\mathcal{Z}}(c) = \min_{w \in \mathcal{Z}} c^\top w$, then

$$\mathbb{E}[C^\top y(X) - v_{\mathcal{Z}}(C)] = \mathcal{E} + \mathbb{E}[v_{\mathcal{Z}}(c_*(X)) - v_{\mathcal{Z}}(C)].$$

The second term is independent of the learned policy and reflects uncertainty remaining after observing the context.

B.3 Optimization Under Sufficient Smoothing

B.3.1 A General PL Transfer Lemma

Lemma B.1 (Polyak–Łojasiewicz chain rule)**.** *Let $K \subseteq \mathbb{R}^d$ be convex and let $U \subseteq \mathbb{R}^d$. Suppose F is differentiable on a neighborhood of K and α-strongly convex on K, with $\alpha > 0$, and let $z_F^\star \in \arg\min_{z\in K} F(z)$. Let p be differentiable on a neighborhood of U, with $p(U) \subseteq K$, and suppose that, for some $\sigma > 0$,*

$$J_p(u)J_p(u)^\top \succeq \sigma^2 I \qquad (u \in U), \tag{33}$$

where $J_p(u) \in \mathbb{R}^{d\times d}$ is the Jacobian of p and $I \in \mathbb{R}^{d\times d}$ is the identity matrix. Then $f = F \circ p$ satisfies the Polyak–Łojasiewicz (PL) inequality

$$\tfrac{1}{2}\|\nabla f(u)\|^2 \geq \alpha\sigma^2\big(f(u) - F(z_F^\star)\big), \qquad u \in U. \tag{34}$$

Proof of Lemma B.1. Strong convexity between $z \in K$ and $z_F^\star$, followed by completing the square, gives

$$F(z) - F(z_F^\star) \leq \nabla F(z)^\top(z - z_F^\star) - \frac{\alpha}{2}\|z - z_F^\star\|^2 \leq \frac{\|\nabla F(z)\|^2}{2\alpha}.$$

The chain rule and Equation (33) give

$$\|\nabla f(u)\|^2 = \nabla F(p(u))^\top J_p(u)J_p(u)^\top \nabla F(p(u)) \geq \sigma^2\|\nabla F(p(u))\|^2.$$

Combining these inequalities proves the claim. If $z_F^\star$ also minimizes F over the full range of p and belongs to $p(U)$, then $F(z_F^\star)$ is the unrestricted minimum of $F \circ p$. □

B.3.2 Coordinates and Radial Geometry

All derivatives in this section are taken in the affine hull. In the original coordinates of Appendix A, let z_∞ be the analytic center, let N have orthonormal columns spanning $\ker E$, and set

$$H_0 = N^\top \nabla^2 V(z_\infty)N, \qquad z = z_\infty + NH_0^{-1/2}y.$$

After subtracting the constant $V(z_\infty)$, the transformed barrier has the form in Equation (21). The transformed cost is $H_0^{-1/2}N^\top c_{\mathrm{orig}}$; thus its Euclidean norm measures cost size in the barrier geometry. In the main text, c_{orig} already denotes the cost in affine-hull coordinates.

The normalization gives

$$\sum_{j=1}^m a_j = 0, \qquad \sum_{j=1}^m a_j a_j^\top = I, \qquad \|a_j\| \leq 1.$$

Consequently, the open unit ball lies in the feasible interior. Write $\mathbb{B}_s = \{w : \|w\| \leq s\}$ and, for a unit direction v, let $\tau(v) = \sup\{t \geq 0 : a_j^\top(tv) \leq 1 \text{ for every } j\}$. Boundedness gives $1 \leq \tau(v) < \infty$, and

$$q(tv) = \min\{t, \tau(v)\}v, \qquad p_\varepsilon(tv) = \psi_v(t)v, \qquad \psi_v(t) = r_\varepsilon(t^2)\min\{t, \tau(v)\}.$$

The radius $\psi_v(t)$ is nondecreasing with t, and $\psi_v(\rho) = \rho_p = \rho r_\varepsilon(\rho^2)$. Since $0 < \rho < 1$, the hard radial map is the identity on a neighborhood of $\mathbb{B}_\rho$. There,

$$p_\varepsilon(u) = r_\varepsilon(\|u\|^2)u, \qquad p_\varepsilon(\mathbb{B}_\rho) = \mathbb{B}_{\rho_p}.$$

The image identity follows because $t \mapsto tr_\varepsilon(t^2)$ is continuous and strictly increasing on $[0, \rho]$, with endpoints 0 and ρ_p.

B.3.3 Proof of Theorem 3

Let $F_{\mu,c}(y) = c^\top y + \mu V(y)$, so that $f_{\mu,c,\varepsilon} = F_{\mu,c} \circ p_\varepsilon$. For a unit vector v, put $\beta_j = a_j^\top v$. Whenever tv is strictly feasible with $t \geq 0$,

$$\frac{d}{dt}V(tv) = \sum_j \frac{\beta_j}{1 - t\beta_j} = t\sum_j \frac{\beta_j^2}{1 - t\beta_j} \geq \frac{t}{1+t}. \tag{35}$$

The equality uses $\sum_j \beta_j = 0$, and the inequality uses $|\beta_j| \le 1$ and $\sum_j \beta_j^2 = 1$. It follows that

$$\frac{d}{dt} F_{\mu,c}(tv) \ge -C_{\max} + \mu \frac{t}{1+t}.$$

Condition (24) makes this derivative nonnegative at $t = \rho_p$ and strictly positive for $t > \rho_p$. The unique interior minimizer $y^\star$ of $F_{\mu,c}$ exists by Appendix A; the radial derivative bound places it in $\mathbb{B}_{\rho_p}$. Since $p_\varepsilon(\mathbb{B}_\rho) = \mathbb{B}_{\rho_p}$, there is a corresponding $u^\star = u^\star_{\mu,c,\varepsilon} \in \mathbb{B}_\rho$ with $p_\varepsilon(u^\star) = y^\star$. Every value of p_ε is strictly feasible, so $F_{\mu,c}(y^\star)$ is the unrestricted minimum of $f_{\mu,c,\varepsilon}$. Hence the output restriction preserves that minimum.

For $y \in \mathbb{B}_{\rho_p}$, the barrier Hessian satisfies

$$\nabla^2 V(y) = \sum_j \frac{a_j a_j^\top}{(1 - a_j^\top y)^2} \succeq \frac{1}{(1+\rho_p)^2} I. \tag{36}$$

On the raw-output ball,

$$J_{p_\varepsilon}(u) = r_\varepsilon(\|u\|^2) I + 2 r'_\varepsilon(\|u\|^2) u u^\top \succeq \varepsilon I.$$

Thus $F_{\mu,c}$ is $\mu/(1+\rho_p)^2$-strongly convex on $K = \mathbb{B}_{\rho_p}$, and $J_{p_\varepsilon}(u) J_{p_\varepsilon}(u)^\top \succeq \varepsilon^2 I$. Lemma B.1, with $U = \mathbb{B}_\rho$, $\alpha = \mu/(1+\rho_p)^2$ and $\sigma = \varepsilon$, proves Equation (25).

Finally, if $t^\star = \|y^\star\| > 0$, stationarity along $v = y^\star/t^\star$ and Equation (35) give

$$0 = c^\top v + \mu \left. \frac{d}{dt} V(tv) \right|_{t=t^\star} \ge -C_{\max} + \mu \frac{t^\star}{1+t^\star}.$$

Since $\mu > C_{\max}$, we obtain $t^\star \le C_{\max}/(\mu - C_{\max})$. The inequality also holds when $y^\star = 0$. As

$$\|y^\star\| = r_\varepsilon(\|u^\star\|^2)\|u^\star\| \ge \varepsilon \|u^\star\|,$$

Equation (26) follows.

C Experimental Details

Algorithm 1 Central Path SPO

Require: Dataset $\mathcal{D}$, constraints (A, b), smoothing parameter $\mu > 0$, learning rate η.

1: Compute the analytic center and interior projection geometry once.
2: Initialize the policy $\pi_\theta = p \circ g_\theta$.
3: **for** each training minibatch $\mathcal{B}$ **do**
4: Evaluate $u_i = g_\theta(x_i)$ and $z_i = p(u_i)$ for $i \in \mathcal{B}$.
5: $\theta \leftarrow \theta - \eta \nabla_\theta \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \big[c_i^\top z_i + \mu V(z_i) \big]$.
6: **end for**
7: For a new context x, predict $\widehat{c} = -\mu \nabla V(\pi_\theta(x))$.
8: Return a solution of the original LP under $\widehat{c}$.

Benchmarks and data. In Table 1, sizes denote grid dimensions for shortest path, variables $\times$ inequality constraints for random LPs, customers $\times$ facilities for Facility Location, and items $\times$ resource constraints for knapsack.

All instances use 100 training context–cost pairs, 32 validation pairs, and 128 test contexts per run. The exact arrays and full seed bundles are matched across all available methods within each instance and run. Learner preparation uses only the observed training data and known optimization structure; validation is used only for configuration and checkpoint selection.

Selection and evaluation. For each method and instance, one seed bundle screens the candidates, followed by a three-seed refinement of the three leading configurations. Validation then fixes the configuration and method-specific training horizon for ten evaluation runs. These runs vary both data and model seeds, with the same bundles used across methods.

Test evaluation uses conditional-mean cost references. Decisions are evaluated on the original benchmark problem. Write $\widetilde{c}_i$ for the evaluation mean cost and $v_i^\star$ for the optimal objective value of that benchmark problem under $\widetilde{c}_i$. The normalized regret for one run is

$$100\,\frac{\sum_{i=1}^{n_{\text{test}}}[\widetilde{c}_i^\top \widehat{z}(x_i) - v_i^\star]}{\sum_{i=1}^{n_{\text{test}}}|v_i^\star|}.$$

Table 1 reports means across runs with sample standard deviations in percentage points. The deviations describe run-to-run variability, not confidence intervals for the means.

Models and hyperparameters. Neural methods use SiLU MLPs and AdamW. Within each instance, all methods share the same joint architecture/optimizer options: nine configurations for knapsack and facility location, and seven for shortest path and random LPs. Method-specific parameters are tuned independently for each method. Knapsack and facility-location checkpoints are at epochs $0, 10, 25, 50, 100, 200, 300$; shortest-path and random-LP checkpoints are at $0, 5, 10, 20, 40, 80, 120, 180, 240, 300$.

Tables 7–13 give the selected architecture, optimizer settings, horizon, and method parameters for every available method–instance pair. For CP-SPO, equalities are eliminated using a tangent basis, and the barrier Hessian at the analytic center whitens the reduced coordinates. The interior projection and its analytic vector–Jacobian product are evaluated in Torch.

Timing protocol. Tables 3 and 4 report optimizer-loop wall and process CPU time on the same Intel Xeon 6730P node. Eight single-thread fits run concurrently, each pinned to a distinct physical core. One prespecified repeat (seed-bundle index 0) replays each validation-selected configuration and horizon. Table 5 reports LAVA's preprocessing time, including adjacent-vertex computation, separately from training. Wall times are not divided by the number of concurrent fits. With one timed repeat, these measurements support coarse comparisons, not timing error bars. A dash denotes unavailable LAVA geometry, not zero runtime. Table 6 reports mean training and preparation times, along with a common-11 training comparison using the same instances for every method.

Table 3: Training wall time (seconds) for the selected configurations and horizons in Table 1. One repeat per setting, with eight concurrent single-thread fits on the same Intel Xeon 6730P node. A dash denotes unavailable LAVA geometry.

Size	CP-SPO	MSE	SPO+	WISE	FY	NCE	LAVA
Shortest path							
10×10	0.790	0.263	2.391	0.043	13.837	0.109	0.553
15×15	0.226	0.265	37.263	0.088	21.254	0.181	6.424
20×20	0.894	0.594	13.587	0.030	69.706	0.166	—
25×25	1.153	0.362	60.981	0.060	215.609	0.246	—
Facility Location							
50×5	0.832	1.012	12.345	0.067	9.578	0.195	1246.938
100×10	4.343	1.460	100.398	0.040	46.968	0.384	—
200×10	11.716	1.278	181.512	0.061	58.833	0.537	—
250×10	17.432	2.167	218.541	0.061	73.971	0.642	—
Random LP							
50×50	2.325	1.711	45.773	1.812	5.221	0.097	2.109
150×50	1.845	0.595	80.882	1.204	38.591	0.108	1.676
200×50	2.286	1.556	139.246	1.252	67.541	0.313	3.915
400×50	2.557	2.237	134.374	2.225	148.573	0.214	26.096
Knapsack							
50×2	0.570	0.377	11.612	0.174	3.194	1.576	2.187
100×2	0.947	0.350	6.441	0.089	3.212	0.162	2.530
200×2	1.177	0.307	29.704	0.109	6.851	0.130	5.553
500×2	1.433	1.281	52.616	0.078	11.210	0.162	59.304

Table 4: Training process CPU time (seconds), measured in the same runs and over the same optimizer-loop interval as Table 3. One repeat per setting; no timing SD is estimated.

Size	CP-SPO	MSE	SPO+	WISE	FY	NCE	LAVA
Shortest path							
10×10	0.779	0.260	2.371	0.042	13.779	0.107	0.551
15×15	0.224	0.262	37.080	0.087	21.178	0.179	6.398
20×20	0.883	0.587	13.513	0.029	69.467	0.165	—
25×25	1.144	0.360	60.651	0.059	214.854	0.244	—
Facility Location							
50×5	0.824	1.000	12.226	0.066	9.518	0.192	1241.433
100×10	4.312	1.443	99.720	0.039	46.553	0.380	—
200×10	11.653	1.265	179.412	0.060	58.616	0.534	—
250×10	17.325	2.142	217.392	0.060	73.693	0.639	—
Random LP							
50×50	2.301	1.696	45.474	1.794	5.188	0.096	2.095
150×50	1.831	0.590	80.417	1.196	38.440	0.107	1.668
200×50	2.266	1.542	138.498	1.240	66.936	0.310	3.897
400×50	2.539	2.218	133.802	2.206	148.013	0.212	25.978
Knapsack							
50×2	0.565	0.372	11.471	0.172	3.165	1.560	2.170
100×2	0.937	0.345	6.386	0.088	3.185	0.160	2.514
200×2	1.166	0.272	29.329	0.108	6.794	0.129	5.521
500×2	1.419	1.263	52.156	0.076	11.128	0.161	59.009

Table 5: LAVA preprocessing wall time (seconds), including adjacent-vertex computation, for the runs in Table 3. Measured separately from training, with one repeat per setting; a dash denotes unavailable LAVA geometry.

Size	LAVA (s)
Shortest path	
10×10	293.307
15×15	827.768
20×20	—
25×25	—
Facility Location	
50×5	492.422
100×10	—
200×10	—
250×10	—
Random LP	
50×50	3.058
150×50	3.859
200×50	3.795
400×50	5.904
Knapsack	
50×2	2.772
100×2	3.203
200×2	3.700
500×2	9.060

Table 6: Arithmetic mean wall times (seconds). Available-setting means use 16 settings except for LAVA's 11; the common-11 training mean uses identical settings for every method. Preparation is reported separately.

Method	Settings	Training	Common 11	Preparation
CP-SPO	16	3.158	1.363	4.352
MSE	16	0.988	0.905	8.917
SPO+	16	70.479	50.241	4.057
WISE	16	0.462	0.649	9.068
FY	16	49.634	29.915	3.781
NCE	16	0.326	0.295	2.792
LAVA	11	123.390	123.390	149.895

Validation-selected configurations. The following tables report the configuration and training horizon selected for each method and instance, held fixed across the ten runs. For CP-SPO, α denotes barrier strength relative to the training-data scale and s denotes the contraction scale.

Table 7: CP-SPO validation-selected configurations. MLP denotes depth $\times$ width. All learning-rate schedules are constant, and each selected barrier schedule has equal initial and final α.

Size	MLP	η	Batch	Epochs	α	Contraction map	s
Shortest path							
10×10	2×64	0.001	100	300	1	hyperbolic	0.3
15×15	3×256	0.003	100	40	1	hyperbolic	1
20×20	3×256	0.0003	100	120	1	hyperbolic	1
25×25	3×256	0.0003	100	120	1	hyperbolic	1
Facility Location							
50×5	2×64	0.001	64	200	3	rational	1
100×10	2×64	0.001	64	300	3	rational	1
200×10	2×64	0.001	64	300	3	rational	1
250×10	2×64	0.001	64	300	3	rational	1
Random LP							
50×50	3×512	0.003	100	300	1	hyperbolic	0.03
150×50	3×512	0.003	100	300	1	hyperbolic	0.03
200×50	3×512	0.001	100	300	1	hyperbolic	0.03
400×50	3×512	0.003	100	300	1	hyperbolic	0.03
Knapsack							
50×2	2×64	0.001	64	200	0.1	hyperbolic	1
100×2	2×64	0.001	64	300	0.1	hyperbolic	1
200×2	2×64	0.001	64	300	0.1	hyperbolic	1
500×2	2×64	0.001	64	200	0.1	hyperbolic	1

Table 8: MSE validation-selected configurations. MLP denotes depth $\times$ width. All learning-rate schedules are constant.

Size	MLP	η	Batch	Epochs
Shortest path				
10×10	3×256	0.003	100	80
15×15	3×256	0.003	100	80
20×20	3×256	0.0003	100	120
25×25	3×256	0.003	100	80
Facility Location				
50×5	3×128	0.0003	64	300
100×10	3×128	0.0003	64	300
200×10	2×64	0.001	32	200
250×10	3×128	0.0003	64	300
Random LP				
50×50	3×512	0.001	100	300
150×50	3×512	0.003	100	120
200×50	3×512	0.001	100	240
400×50	3×512	0.001	100	300
Knapsack				
50×2	2×64	0.001	32	100
100×2	3×128	0.003	64	100
200×2	3×128	0.003	64	100
500×2	3×128	0.001	64	300

Table 9: SPO+ validation-selected configurations. MLP denotes depth $\times$ width. All learning-rate schedules are constant.

Size	MLP	η	Batch	Epochs
Shortest path				
10×10	2×64	0.003	100	20
15×15	3×256	0.0003	100	180
20×20	3×256	0.0003	100	40
25×25	3×256	0.0003	100	120
Facility Location				
50×5	3×128	0.001	64	100
100×10	3×128	0.001	64	300
200×10	3×128	0.001	64	300
250×10	2×64	0.001	32	300
Random LP				
50×50	3×512	0.0003	100	240
150×50	3×512	0.0003	100	180
200×50	2×64	0.003	100	240
400×50	3×512	0.0003	100	120
Knapsack				
50×2	3×128	0.001	64	200
100×2	3×128	0.001	64	100
200×2	3×128	0.001	64	300
500×2	3×128	0.0003	64	300

Table 10: WISE validation-selected configurations. MLP denotes depth $\times$ width. All learning-rate schedules are constant.

Size	MLP	η	Batch	Epochs
Shortest path				
10×10	3×256	0.0003	100	10
15×15	3×256	0.0003	100	20
20×20	3×256	0.0003	100	5
25×25	3×256	0.0003	100	10
Facility Location				
50×5	2×64	0.0003	64	25
100×10	2×64	0.0003	64	10
200×10	2×64	0.0003	64	10
250×10	2×64	0.0003	64	10
Random LP				
50×50	3×512	0.0003	100	300
150×50	3×512	0.001	100	240
200×50	3×512	0.001	100	180
400×50	3×512	0.0003	100	300
Knapsack				
50×2	2×64	0.001	32	50
100×2	3×128	0.0003	64	25
200×2	2×64	0.0003	64	50
500×2	3×128	0.0003	64	25

Table 11: FY validation-selected configurations. MLP denotes depth $\times$ width. All learning-rate schedules are constant; FY uses one perturbation per example.

Size	MLP	η	Batch	Epochs	σ
Shortest path					
10×10	2×64	0.003	100	240	10.0
15×15	2×64	0.003	100	120	10.0
20×20	2×64	0.003	100	180	10.0
25×25	2×64	0.0003	100	240	1.0
Facility Location					
50×5	3×128	0.001	64	200	10.0
100×10	2×64	0.003	64	300	10.0
200×10	3×128	0.001	64	200	10.0
250×10	3×128	0.001	64	200	10.0
Random LP					
50×50	3×512	0.0003	100	80	0.1
150×50	3×512	0.001	100	240	10.0
200×50	3×512	0.0003	100	300	1.0
400×50	3×512	0.0003	100	300	1.0
Knapsack					
50×2	2×64	0.001	32	200	1.0
100×2	3×128	0.003	64	200	1.0
200×2	2×64	0.001	32	300	10.0
500×2	2×64	0.001	32	300	10.0

Table 12: NCE validation-selected configurations. MLP denotes depth × width.

Size	MLP	η	Batch	Epochs	Schedule	Solve ratio
Shortest path						
10×10	2×64	0.003	100	10	constant	0.2
15×15	3×256	0.0003	100	20	constant	0.05
20×20	3×256	0.001	100	5	constant	0.05
25×25	2×64	0.001	100	20	cosine	0.05
Facility Location						
50×5	2×64	0.003	64	25	constant	0.05
100×10	2×64	0.003	64	25	constant	0.05
200×10	2×64	0.003	64	25	constant	0.05
250×10	2×64	0.003	64	25	constant	0.05
Random LP						
50×50	3×512	0.001	100	5	constant	0.2
150×50	3×512	0.001	100	5	constant	0.2
200×50	3×512	0.0003	100	10	constant	0.2
400×50	3×512	0.001	100	5	constant	0.2
Knapsack						
50×2	2×64	0.001	100	300	constant	0.2
100×2	2×64	0.003	64	10	constant	1.0
200×2	2×64	0.001	64	25	constant	0.05
500×2	2×64	0.001	64	25	constant	0.05

Table 13: LAVA validation-selected configurations. MLP denotes depth × width. All learning-rate schedules are constant; vertex tolerance is zero.

Size	MLP	η	Batch	Epochs	Threshold
Shortest path					
10×10	3×256	0.003	100	20	-1.0
15×15	3×256	0.003	100	40	-1.0
Facility Location					
50×5	3×128	0.001	64	300	-1.0
Random LP					
50×50	3×512	0.0003	100	180	-0.1
150×50	3×512	0.001	100	80	-0.1
200×50	3×512	0.003	100	120	-0.1
400×50	3×512	0.003	100	180	-1.0
Knapsack					
50×2	3×128	0.001	64	300	-1.0
100×2	3×128	0.001	64	200	-1.0
200×2	3×128	0.003	64	200	-1.0
500×2	3×128	0.001	64	300	-1.0

D HYPERPARAMETER SENSITIVITY

We vary barrier strength, learning rate, and contraction scale separately over 0.1, 0.3, 1, 3, and 10 times each instance's validation-selected value. Architecture, data, seeds, all other parameters, and the selected training horizon remain fixed. The $1\times$ column reuses the main results. Barrier-strength changes scale both endpoints of the barrier schedule; contraction-scale changes also modify the implementation-derived output bound. Every grid point is reported, and none of these test results changes the main selection.

Barrier strength has little effect across the tested range on shortest path and knapsack, but matters substantially on facility location. For example, at 100×10, regret rises from 16.61% at the selected strength to 47.54% and 53.87% at $0.1\times$ and $10\times$. On random LPs, the largest barrier multiplier also increases regret. Learning rate has a pronounced effect on shortest path and knapsack at these fixed horizons. Contraction scale is comparatively stable on facility location and random LPs, but can matter on smaller knapsack instances: at 50×2, the $10\times$ multiplier raises regret from 9.92% to 16.53%.

Table 14: Barrier-strength sensitivity: normalized expected test regret (%), mean with sample SD on the line below, over ten matched runs. Multipliers apply to each instance's validation-selected value; 1× reuses the main results. All other settings and the selected training horizon are fixed.

Size	Selected α	0.1×	0.3×	1×	3×	10×
Shortest path						
		8.28	8.30	8.25	8.25	8.79
10×10	1	±1.17	±1.17	±1.19	±1.11	±1.08
		7.93	7.95	8.04	7.92	9.14
15×15	1	±1.43	±1.41	±1.40	±1.33	±1.38
		7.99	7.98	7.97	7.88	8.03
20×20	1	±0.71	±0.72	±0.71	±0.70	±0.83
		7.97	7.95	7.90	7.80	8.11
25×25	1	±0.73	±0.72	±0.74	±0.72	±0.99
Facility Location						
		25.59	15.43	10.76	17.52	28.55
50×5	3	±1.83	±1.44	±1.04	±1.41	±1.85
		47.54	25.79	16.61	30.93	53.87
100×10	3	±3.59	±1.80	±0.60	±2.02	±3.10
		61.40	31.74	18.96	33.84	62.11
200×10	3	±3.73	±1.45	±1.27	±1.83	±1.91
		68.01	34.05	20.58	38.13	67.86
250×10	3	±3.84	±2.10	±1.45	±2.94	±2.55
Random LP						
		2.01	2.04	2.01	2.12	4.44
50×50	1	±0.59	±0.61	±0.60	±0.63	±1.31
		1.43	1.42	1.43	1.46	3.20
150×50	1	±0.45	±0.45	±0.47	±0.45	±0.58
		1.52	1.52	1.51	1.51	2.99
200×50	1	±0.40	±0.40	±0.39	±0.39	±0.41
		1.21	1.21	1.22	1.23	2.43
400×50	1	±0.31	±0.31	±0.32	±0.32	±0.64
Knapsack						
		9.85	9.94	9.92	9.86	9.85
50×2	0.1	±1.29	±1.38	±1.43	±1.61	±1.65
		9.16	9.15	9.15	9.14	9.25
100×2	0.1	±1.14	±1.16	±1.15	±1.24	±1.23
		9.54	9.53	9.53	9.53	9.60
200×2	0.1	±0.49	±0.48	±0.48	±0.47	±0.47
		9.17	9.17	9.18	9.17	9.19
500×2	0.1	±0.71	±0.71	±0.71	±0.71	±0.74

Table 15: Learning-rate sensitivity: normalized expected test regret (%), mean with sample SD on the line below, over ten matched runs. Multipliers apply to each instance's validation-selected value; $1\times$ reuses the main results. All other settings and the selected training horizon are fixed.

Size	Selected η	$0.1\times$	$0.3\times$	$1\times$	$3\times$	$10\times$
Shortest path						
		17.40	9.74	8.25	10.48	15.23
10×10	0.001	± 1.54	± 1.11	± 1.19	± 1.43	± 2.04
		17.99	8.70	8.04	9.42	14.46
15×15	0.003	± 2.08	± 0.91	± 1.40	± 1.59	± 2.21
		30.80	12.15	7.97	7.98	11.29
20×20	0.0003	± 2.86	± 1.09	± 0.71	± 0.91	± 0.99
		28.69	11.65	7.90	9.01	11.95
25×25	0.0003	± 4.57	± 1.41	± 0.74	± 1.14	± 1.09
Facility Location						
		22.04	11.34	10.76	12.67	15.84
50×5	0.001	± 1.84	± 1.01	± 1.04	± 1.27	± 1.54
		22.80	16.88	16.61	19.95	20.03
100×10	0.001	± 1.71	± 1.09	± 0.60	± 1.42	± 1.41
		25.64	19.37	18.96	22.07	21.96
200×10	0.001	± 1.27	± 1.04	± 1.27	± 1.37	± 1.81
		26.95	20.90	20.58	23.82	22.95
250×10	0.001	± 1.44	± 1.11	± 1.45	± 1.34	± 1.63
Random LP						
		2.69	2.11	2.01	2.02	1.94
50×50	0.003	± 0.77	± 0.64	± 0.60	± 0.51	± 0.32
		1.77	1.49	1.43	1.42	1.31
150×50	0.003	± 0.48	± 0.48	± 0.47	± 0.46	± 0.45
		2.27	1.72	1.51	1.43	1.36
200×50	0.001	± 0.60	± 0.44	± 0.39	± 0.40	± 0.39
		1.45	1.30	1.22	1.39	1.15
400×50	0.003	± 0.45	± 0.32	± 0.32	± 0.40	± 0.45
Knapsack						
		27.66	13.48	9.92	8.94	9.01
50×2	0.001	± 3.50	± 2.87	± 1.43	± 0.87	± 1.07
		16.31	11.11	9.15	8.81	9.75
100×2	0.001	± 1.74	± 1.47	± 1.15	± 1.02	± 0.70
		14.80	10.74	9.53	9.58	10.13
200×2	0.001	± 1.06	± 0.68	± 0.48	± 0.70	± 0.71
		15.73	10.84	9.18	9.18	9.98
500×2	0.001	± 1.07	± 0.91	± 0.71	± 0.77	± 0.67

Table 16: Contraction-scale sensitivity: normalized expected test regret (%), mean with sample SD on the line below, over ten matched runs. Multipliers apply to each instance's validation-selected value; $1\times$ reuses the main results. All other settings and the selected training horizon are fixed.

Size	Selected s	$0.1\times$	$0.3\times$	$1\times$	$3\times$	$10\times$
Shortest path						
		9.40	9.17	8.25	8.05	7.65
10×10	0.3	± 1.51	± 1.43	± 1.19	± 0.99	± 0.92
		8.89	8.03	8.04	7.68	8.92
15×15	1	± 1.46	± 1.34	± 1.40	± 1.39	± 1.53
		8.14	7.82	7.97	8.02	8.89
20×20	1	± 0.59	± 0.67	± 0.71	± 0.79	± 0.68
		9.06	8.39	7.90	8.23	8.71
25×25	1	± 1.25	± 1.04	± 0.74	± 0.67	± 0.91
Facility Location						
		10.75	10.80	10.76	10.69	10.33
50×5	1	± 0.92	± 0.99	± 1.04	± 0.94	± 0.80
		16.58	16.79	16.61	16.51	16.25
100×10	1	± 0.80	± 0.74	± 0.60	± 0.60	± 0.60
		18.69	18.96	18.96	18.35	18.11
200×10	1	± 0.81	± 1.29	± 1.27	± 0.91	± 0.68
		20.75	20.84	20.58	20.35	20.22
250×10	1	± 1.68	± 1.23	± 1.45	± 1.84	± 1.29
Random LP						
		2.02	2.02	2.01	1.96	2.19
50×50	0.03	± 0.61	± 0.61	± 0.60	± 0.58	± 0.58
		1.44	1.44	1.43	1.39	1.53
150×50	0.03	± 0.47	± 0.47	± 0.47	± 0.47	± 0.52
		1.51	1.51	1.51	1.49	1.60
200×50	0.03	± 0.39	± 0.39	± 0.39	± 0.37	± 0.45
		1.22	1.22	1.22	1.22	1.29
400×50	0.03	± 0.32	± 0.32	± 0.32	± 0.32	± 0.29
Knapsack						
		8.70	9.16	9.92	11.28	16.53
50×2	1	± 1.12	± 1.54	± 1.43	± 1.79	± 2.15
		8.80	8.88	9.15	9.85	11.12
100×2	1	± 1.12	± 1.12	± 1.15	± 1.42	± 1.36
		9.53	9.51	9.53	9.56	10.19
200×2	1	± 0.53	± 0.53	± 0.48	± 0.42	± 0.53
		9.46	9.31	9.18	9.36	9.76
500×2	1	± 0.75	± 0.79	± 0.71	± 0.85	± 0.91

E Direct-Policy Learning Ablations

We compare CP-SPO with direct-policy learning using Soft-Radial (Schneider & Kuhn, 2026) and RAYEN (Tordesillas et al., 2023) feasibility maps. The ablation uses the largest pure LP of each continuous family: shortest path 25×25 and random LP 400×50, each with 100 training pairs, 32 validation pairs, and 128 test contexts. Data and full seed bundles are matched to CP-SPO. Each direct policy minimizes the observed linear objective, without a barrier penalty, and deploys its feasible output directly. It neither recovers a cost nor solves a final LP. The policies are trained on context–cost pairs, not on optimal decisions.

Within this ablation, selection uses unsmoothed validation regret, followed by three-seed refinement and ten matched evaluation runs.

Table 17: Direct-policy comparison: normalized expected test regret (%), mean with sample SD on the line below, over ten matched runs, with 100 training pairs, 32 validation pairs, and 128 test contexts. Parameterization, loss, and deployment change jointly.

Problem	Size	CP-SPO	RAYEN	Soft-Radial
		7.90	88.81	88.91
Shortest path	25×25	± 0.74	± 4.41	± 4.52
		1.22	47.19	46.40
Random LP	400×50	± 0.32	± 2.08	± 2.19

Both direct-policy variants have substantially higher regret in these two settings. Soft-Radial and RAYEN give 88.91% and 88.81% on shortest path, compared with 7.90% for CP-SPO; on the random LP, they give 46.40% and 47.19%, compared with 1.22%. These results support learning the smoothed interior policy and recovering its cost on the tested instances. Parameterization, loss, and deployment change jointly, so the comparison does not isolate the effect of smoothing. These ablations were not timed.