DECISION-FOCUSED ON-POLICY LEARNING FOR CONTEXTUAL LINEAR OPTIMIZATION WITH PARTIAL FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-focused learning (DFL) trains predictive models by optimizing downstream decision quality rather than standalone prediction accuracy. For contextual linear optimization, most existing DFL methods assume offline data and full observations of the objective cost vector. We develop an on-policy learning method for sequential contextual linear optimization under partial feedback, generalizing the standard bandit feedback setting. Our method learns a stochastic predict-then-optimize policy that samples a cost-vector prediction from a conditional distribution and solves the resulting downstream linear optimization problem. To update this distributional model, we introduce a two-component hybrid gradient estimator. The first component is a score function estimator, which provides an unbiased but potentially high-variance policy gradient estimate. The second is a decision-focused plug-in component that uses an auxiliary nuisance estimate of the latent cost vector to exploit the downstream optimization structure, becoming more informative as the estimate improves. We prove an $\mathcal{O}(T^{-1/2})$ bound on the average squared policy-gradient norm, matching the standard non-convex SGD rate. Experiments on top-k selection, shortest path, combinatorial pricing, and a real-data energy-scheduling benchmark show that, under bandit feedback, the hybrid gradient approach achieves lower mean cumulative regret than the contextual-bandit baselines on all four benchmarks and than linear Thompson sampling on three, and that it also works with richer conditional generative cost models.

1 INTRODUCTION

Many operational systems make sequential decisions by solving constrained optimization problems given observed contextual information, with applications spanning vehicle routing, dynamic pricing, power scheduling, product allocation, order fulfillment, and recommendation (Van Hentenryck & Bent, 2006; Powell, 2022; Elmachtoub & Grigas, 2022; Li et al., 2010; Zhao et al., 2026; Sadana et al., 2025; Ye et al., 2026b). In these settings, the feasible set and operational constraints are known, while the relevant cost or reward parameters are uncertain at decision time and instead must be learned. When such systems are deployed repeatedly, the learner is not merely trained on a fixed historical dataset; its own decisions become part of the data-generating process, determining which outcomes are observed and hence what information is available for future learning. Furthermore, the collected data rarely include full information about problem parameters: a selected route reveals only the realized travel time, a posted price reveals only demand at that price, a recommended subset reveals only response on the chosen items. The learner thus repeatedly sees a context, chooses a feasible decision from the current policy, and observes only the feedback generated by that decision. This paper studies sequential contextual linear optimization (CLO), a widely applicable class of constrained optimization problems, in this on-policy decision-making environment with partial feedback.

The combination of constrained optimization and online decision-making with partial feedback is not yet captured systematically by any one existing framework. Decision-focused learning (DFL) trains predictions through the decisions they induce (Mandi et al., 2024; Elmachtoub & Grigas, 2022), but most existing methods assume batch data and full observations of the latent cost vector in CLO. Contextual bandits and online learning study sequential learning from partial feedback, including

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.

large and structured action spaces (Li et al., 2010; Abbasi-Yadkori et al., 2011; Krishnamurthy et al., 2016; Zhu et al., 2022), but typically model rewards directly over context-action pairs rather than preserving and strongly exploiting the downstream linear optimization model. Given the surge of development in powerful DFL methods for offline problems with full data (Mandi et al., 2024), a natural question emerges: can DFL methods be adapted to on-policy, sequential decision-making with partial feedback?

We answer affirmatively by introducing a new class of on-policy stochastic gradient estimators that enables strong plug-in/predict-then-optimize policies. We learn a structured conditional distribution over cost vectors and induce decisions through linear optimization, rather than a direct context-to-decision map. We update the policy parameters by mixing two gradient signals with complementary trade-offs. A score function estimator uses the on-policy realized cost directly, making it unbiased for the induced policy objective and naturally tied to exploration through sampling. However, the score function approach treats the downstream optimizer as a black-box action generator and therefore does not strongly use the structure of the feasible set or linear objective in its gradient. We therefore propose a decision-focused plug-in gradient estimator that uses this structure: it pairs the known feasible set with a nuisance estimate of the conditional mean cost vector, and extends the plug-in approach that Hu et al. (2024b; 2025) developed for offline, off-policy data to on-policy learning. Once the nuisance estimate is plugged-in to a cost approximation, the resulting problem is equivalent to a full information contextual stochastic linear optimization problem and can be tackled with all of the machinery, such as surrogate loss functions, developed in recent years (Mandi et al., 2024; Sadana et al., 2025). We bound the bias of this nuisance estimator approach and show its advantages grow as data accumulates.

Contributions. *First*, we develop an on-policy decision-focused learning method for online contextual linear optimization under a generic partial-feedback model that subsumes bandit, semi-bandit, and full-information settings. The policy update is a hybrid gradient combining an unbiased score function term with a decision-focused plug-in term from a learned auxiliary nuisance cost-vector model. We extend the decision-focused plug-in term to accommodate a broad class of DFL surrogates (e.g., the Smart Predict-then-Optimize surrogate (SPO+), Perturbation Gradient (PG), and Perturbed Fenchel–Young loss (PFYL)). *Second*, we exploit a universal-minimizer property of the nuisance least-squares objective to analyze the nuisance update as standalone online convex optimization against a fixed comparator, independently of the policy iterates. We then convert the resulting regret into a bias bound on the plug-in gradient via a one-step policy descent inequality, yielding $\mathcal{O}(T^{-1/2})$ stationarity. The same argument extends to the surrogate variant. *Third*, numerical experiments on top-k selection, shortest path, combinatorial pricing, and a real-data energy-scheduling benchmark show, under bandit feedback, lower mean cumulative regret than contextual-bandit baselines on all benchmarks and than linear Thompson sampling on three of four.

1.1 Literature review

Decision-focused and predict-then-optimize learning. Decision-focused learning trains predictive models through the downstream optimization problems that consume their outputs (Mandi et al., 2024). For linear objectives, Smart Predict-then-Optimize (SPO) and SPO+ give a task-aware alternative to prediction error (Elmachtoub & Grigas, 2022). Later methods develop differentiable perturbation-based optimizers and directional-gradient estimators (Berthet et al., 2020; Gupta & Huang, 2024). Score function estimators extend DFL to cases where direct differentiation through the task loss or optimization layer is difficult (Silvestri et al., 2026), and distributional DFL methods model uncertainty in optimization parameters using generative predictors such as conditional normalizing flows and diffusion models (Wang et al., 2026; Zhao et al., 2026). Our proposed method inherits the use of distributional predictions and score function gradients, but the observation model differs: the learning signal is online, with policy-generated partial feedback rather than observed cost vectors. Sequential DFL methods include online predict-then-optimize (Liu & Grigas, 2022) and multi-stage Predict+Optimize with sequential parameter revelation and interleaved decisions (Hu et al., 2024a); the closest theoretical analog is online DFL for linear optimization over polytopes (Capitaine et al., 2026). All three share our sequential viewpoint but assume full-information feedback rather than the partial feedback induced by the implemented decision.

Contextual optimization with partial feedback. The closest contextual optimization precursor is Hu et al. (2024b), who study offline contextual linear optimization under partial feedback (pure bandit and semi-bandit). Their work develops, from logged partial-feedback data, induced empirical risk minimization and surrogate approaches using least-squares nuisance estimation. Other work studies partial information from different angles. Ye et al. (2026a) use distributional predictions for mixed-integer and nonlinear contextual stochastic optimization. Bennouna et al. (2025; 2026) study data informativeness for linear optimization, characterizing when partial observations contain enough information to identify an optimal decision. Bouvier et al. (2025) learn randomized combinatorial-optimization policies offline, from full observations of the uncertainty, through a primal-dual Fenchel-Young surrogate. This paper studies the online counterpart. At each iteration, the current policy and optimization oracle choose a feasible decision. Feedback is observed only for that decision, from aggregate loss in the pure bandit case to selected component observations in semi-bandit variants. The same stream of partial feedback is used both to estimate the cost model and to update the policy.

Contextual bandits and policy optimization. Contextual bandits provide the standard framework for sequential learning from partial feedback (Li et al., 2010; Abbasi-Yadkori et al., 2011; Bubeck & Cesa-Bianchi, 2012). Work on large or structured action spaces includes contextual semi-bandits and oracle-based methods (Krishnamurthy et al., 2016; Zhu et al., 2022; Zierahn et al., 2023). Structured prediction with bandit feedback is also related: a learner outputs a structured object and observes only its task loss (Sokolov et al., 2016; Kreutzer et al., 2017). Our score function estimator follows the likelihood-ratio principle of REINFORCE and policy-gradient methods (Williams, 1992; Sutton et al., 2000; Konda & Tsitsiklis, 2000). Unlike standard contextual-bandit models that parameterize rewards over context-action pairs, we keep the linear-cost representation explicit, sampling a cost-vector prediction and solving a known optimization problem, while the auxiliary estimator learns conditional cost structure from partial observations.

Online convex optimization. Online convex optimization (OCO) and its bandit variants study sequential convex or linear losses with full-information or bandit feedback (Zinkevich, 2003; Flaxman et al., 2005; Dani et al., 2008; Abernethy et al., 2008; Ito et al., 2019), bounding regret against the best fixed decision in hindsight. Our setting is one step removed: we update parameters of a stochastic policy that produces a decision via the optimization oracle, and because the induced policy objective is nonsmooth and non-convex, our analysis targets first-order stationarity rather than convex regret.

2 Decision-focused on-policy learning with partial feedback

We consider an online contextual stochastic linear optimization setting. At each time $t = 1, 2, \ldots$, the decision-maker observes a context $x_t \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ (context dimension d_x) and then makes a decision $w_t \in \mathcal{W}$, where $\mathcal{W} \subset \mathbb{R}^{d_w}$ is a known compact feasible set, fixed across contexts. The context x_t is drawn from an unknown marginal distribution $\mathcal{D}_x$. Furthermore, the latent cost vector c_t, drawn from an unknown conditional distribution $p^\star(\cdot \mid x_t)$, is never observed directly: once a decision is made, the learner sees only a partial-feedback vector $v_t \in \mathbb{R}^{d_v}$ (with feedback dimension d_v) whose form depends on the implemented decision; for instance, a selected route reveals the realized travel time of the chosen path, not the time on every edge. We denote the unknown joint distribution of $(x_t, c_t) \in \mathcal{X} \times \mathbb{R}^{d_w}$ by $\mathcal{D}$ (assumed to be i.i.d. over time).

Assumption 1 (Feedback structure)**.** *At each time $t = 1, 2, \ldots$, given that $(x_t, c_t) \sim \mathcal{D}$, we have the following:*

1. *For every possible decision $w_t \in \mathcal{W}$, there exists a known observation matrix $H(w_t) \in \mathbb{R}^{d_v \times d_w}$ such that the observed feedback $v_t \in \mathbb{R}^{d_v}$ satisfies $v_t = H(w_t)c_t$.*
2. *For every possible decision $w_t \in \mathcal{W}$, there exists a known linkage vector $e(w_t) \in \mathbb{R}^{d_v}$ such that $H(w_t)^\top e(w_t) = w_t$; consequently, $y_t = c_t^\top w_t = e(w_t)^\top v_t$ is always recoverable.*

In words, $H(w_t)$ selects which linear measurements of the latent cost vector are revealed by decision w_t, and $e(w_t)$ is a known weight vector whose inner product with the observation always returns the realized scalar cost: $e(w_t)^\top v_t = c_t^\top w_t = y_t$, regardless of the feedback regime. For example, pure bandit feedback ($H(w) = w^\top$, $e(w) = 1$) is the trivial case where v_t is already that scalar cost; Appendix A instantiates the full-information and semi-bandit cases.

In contextual linear optimization, a key assumption is that we can readily solve linear optimization problems over the compact decision set $\mathcal{W}$; let $w^\star(\hat{c}) \in \arg\min_{w\in\mathcal{W}} \hat{c}^\top w$ denote a particular linear optimization oracle for $\mathcal{W}$. We consider a class of distributional plug-in/predict-then-optimize policies, whereby $p_\theta(\cdot \mid x)$ denotes a parametric conditional model for cost vector predictions, parameterized by $\theta \in \mathbb{R}^{d_\theta}$. Any $\theta \in \mathbb{R}^{d_\theta}$ induces a policy $\pi_\theta : \mathcal{X} \to \Delta(\mathcal{W})$, where $\Delta(\mathcal{W})$ is the set of all distributions supported on $\mathcal{W}$. That is, given the context $x_t \in \mathcal{X}$ at time t, we first sample a predicted cost vector $\hat{c}_t \sim p_\theta(\cdot \mid x_t)$ and then make the decision $w_t = w^\star(\hat{c}_t)$. Given $\theta \in \mathbb{R}^{d_\theta}$, let us also define the mean prediction function $\bar{\mu}_\theta(x) := \mathbb{E}_{\hat{c}\sim p_\theta(\cdot|x)}[\hat{c}]$ and the mean decision function $w^\star_\theta(x) := \mathbb{E}_{\hat{c}\sim p_\theta(\cdot|x)}[w^\star(\hat{c})]$. Furthermore, for a given parameter θ, let $\mathcal{D}_\theta$ denote the distribution over (x, c, w) induced by $(x, c) \sim \mathcal{D}$ and $w \sim \pi_\theta(\cdot \mid x)$. An important observation is that the conditional independence property $w \perp c \mid x$ holds for $(x, c, w) \sim \mathcal{D}_\theta$.

Let $f^\star(x) := \mathbb{E}_{c\sim p^\star(\cdot|x)}[c]$ denote the ground truth regression function. We define the expected cost of policy parameterized by $\theta \in \mathbb{R}^{d_\theta}$ by

$$V^{\mathrm{cost}}(\theta) := \mathbb{E}_{(x,c,w)\sim\mathcal{D}_\theta}\left[c^\top w\right] = \mathbb{E}_{x\sim\mathcal{D}_x}\left[\mathbb{E}_{c\sim p^\star(\cdot|x)}\left[c^\top w^\star_\theta(x)\right]\right] = \mathbb{E}_{x\sim\mathcal{D}_x}\left[f^\star(x)^\top w^\star_\theta(x)\right],$$

where the latter equalities use $w \perp c \mid x$. It is well known and apparent from the above that, under full information, the decision maker would ideally make decision $w^\star(f^\star(x))$. Therefore, any policy such that $w^\star_\theta(x) = w^\star(f^\star(x))$ almost surely over x is an optimal full information policy, for example a degenerate distribution with $\bar{\mu}_\theta(x) = f^\star(x)$. The distributional class serves two purposes: a usable gradient and exploration. With a point prediction, the decision $w^\star(\bar{\mu}_\theta(x))$ is piecewise constant in θ, so its gradient is zero almost everywhere. Sampling $\hat{c}$ from $p_\theta(\cdot \mid x)$ smooths this map, and the realized cost y_t alone then gives an unbiased gradient. Offline surrogates such as SPO+ and PFYL avoid the zero gradient by using the observed cost vector. Under partial feedback this vector is not observed, so the plug-in route below first estimates it with a nuisance model. Sampling in cost space also explores through the same oracle.

2.1 Score function gradient estimation

The smoothed map $w^\star_\theta(x) = \mathbb{E}_{\hat{c}\sim p_\theta(\cdot|x)}[w^\star(\hat{c})]$ is a perturbed optimizer in the sense of Berthet et al. (2020), used here as a randomized policy whose sampled decision is implemented each round. Appendix B gives its regularity properties. Since θ enters $V^{\mathrm{cost}}(\theta)$ only through $p_\theta(\cdot \mid x)$, not through the cost or the optimizer $w^\star(\cdot)$, its gradient can be estimated from bandit feedback without differentiating through $w^\star$.

Full details of the score function gradient estimator derivation are given in Appendix D; in short, given current parameters θ_t at time t, under well-behaved parametric distributions, differentiating under the expectation yields the one-sample score function gradient estimator[1]

$$g_t^{\mathrm{score}} := y_t \nabla_\theta \log p_{\theta_t}(\hat{c}_t \mid x_t), \quad \text{where} \quad \hat{c}_t \sim p_{\theta_t}(\cdot \mid x_t), w_t = w^\star(\hat{c}_t), y_t = c_t^\top w_t, \tag{1}$$

which satisfies $\mathbb{E}[g_t^{\mathrm{score}}] = \nabla_\theta V^{\mathrm{cost}}(\theta_t)$. Note that g_t^{score} depends on the environment only through the scalar bandit feedback y_t, with no explicit reference to $f^\star$. More generally, any terms preserving unbiasedness of the gradient estimator of $\nabla_\theta V^{\mathrm{cost}}(\theta)$ may replace y_t in (1); for instance, subtracting a context-dependent baseline $b(x_t)$ from y_t preserves unbiasedness while typically reducing variance.

2.2 Decision-focused gradient estimation

The score function gradient estimator, while unbiased, is structure-agnostic: it treats y_t as a generic scalar cost and only weakly exploits the linear optimization structure. Decision-focused methods in the full-information setting instead exploit the direct dependence of $w^\star_\theta(\cdot)$ on θ. Furthermore, particularly in the case of contextual linear optimization, the decomposition of V^{cost} above reveals that $V^{\mathrm{cost}}(\theta)$ depends on the cost distribution *only* through its conditional mean $f^\star(x) = \mathbb{E}[c \mid x]$. At the same time, the observed feedback v_t carries information about $f^\star$ that the score estimator never directly exploits. These observations suggest a complementary route: estimate $f^\star$ from accumulated feedback, plug the resulting nuisance estimate into $V^{\mathrm{cost}}(\cdot)$, and apply standard full-information decision-focused machinery such as direct gradients and surrogate losses. This decision-focused, plug-in perspective trades the unbiasedness of (1) for exploitation of the downstream optimization problem and a potentially substantial reduction in variance.

[1] "Score function" refers to the likelihood-ratio gradient $\nabla_\theta \log p_\theta$ of REINFORCE (Williams, 1992). Hu et al. (2024b) use the term differently, for pseudo-cost vectors.

Nuisance estimation via least-squares feedback fitting. We introduce an auxiliary cost-vector regression model $f_\phi : \mathcal{X} \to \mathbb{R}^{d_w}$ (the *nuisance* estimator), parameterized by $\phi \in \Phi \subset \mathbb{R}^{d_\phi}$, which approximates the conditional cost mean $f^\star(x) = \mathbb{E}[c \mid x]$ by minimizing the following least-squares objective based on the observed feedback $v = H(w)c$.

$$\min_{\phi \in \Phi} \mathcal{R}_\theta(f_\phi), \qquad \mathcal{R}_\theta(f_\phi) := \mathbb{E}_{(x,c,w)\sim\mathcal{D}_\theta}\left[\|v - H(w) f_\phi(x)\|^2\right] \tag{2}$$

Remark 2.1 (Universal minimizer)**.** *A key property of this objective is that $f^\star$ is a* universal *minimizer of $\mathcal{R}_\theta$ for every $\theta \in \mathbb{R}^{d_\theta}$. Consequently, the nuisance estimation target does not shift with the policy, which is central to our analysis.*

The regret analysis of the nuisance model is obtained using standard online convex optimization results and universality of the minimizer. The nuisance update is the projected stochastic gradient descent (SGD) step $\phi_{t+1} = \Pi_\Phi(\phi_t - \eta_t \nabla_\phi \ell_t(\phi_t))$ on $\ell_t(\phi) = \|v_t - H(w_t) f_\phi(x_t)\|^2$, with Π_Φ the Euclidean projection onto Φ. We defer the full analysis of the nuisance estimator to Appendix C.

Proposition 2.1 (Nuisance regret)**.** *Suppose Φ is convex and compact with diameter D_Φ, ℓ_t has $G_\mathcal{R}$-bounded gradients and $\nabla^2_\phi \mathcal{R}_\theta(f_\phi) \succeq \lambda I$ for every θ. Under realizability ($f_{\phi^\star} = f^\star$ for some $\phi^\star \in \Phi$), for any sequence $\{\theta_t\}_{t=1}^T$, the average excess nuisance loss satisfies*

(i) $\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f^\star)] \le \frac{D_\Phi G_\mathcal{R}}{\sqrt{T}}$ *with $\eta_t = D_\Phi/(G_\mathcal{R}\sqrt{T})$;*

(ii) $\frac{1}{T}\sum_{t=1}^T \mathbb{E}[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f^\star)] \le \frac{G_\mathcal{R}^2(1+\log T)}{2\lambda T}$ *with $\eta_t = 1/(\lambda t)$.*

Three conditions drive this result. First, the target $f^\star$ is the same for every policy (Remark 2.1), so the nuisance model tracks a fixed target while the data-generating policy changes. Second, realizability ($f^\star \in \mathcal{F} := \{f_\phi : \phi \in \Phi\}$, Assumption 5) puts this target in the nuisance class, so the plug-in gradient has no approximation error. Third, the uniform strong-convexity condition $\nabla^2_\phi \mathcal{R}_\theta(f_\phi) \succeq \lambda I$ for every θ is an identifiability/exploration assumption on the policy-induced feedback; under pure bandit feedback it requires the sampled actions to span the cost directions relevant to the nuisance class. This regret bound is the primary tool for controlling the bias of the plug-in gradient estimator introduced next. Given a nuisance estimate f_ϕ, we define the plug-in policy objective, which is an approximation of $V^{\mathrm{cost}}(\theta)$, by

$$J(\phi, \theta) := \mathbb{E}_{x\sim\mathcal{D}_x}\left[f_\phi(x)^\top w^\star_\theta(x)\right] = \mathbb{E}_{x\sim\mathcal{D}_x, \hat{c}\sim p_\theta(\cdot|x)}\left[f_\phi(x)^\top w^\star(\hat{c})\right]. \tag{3}$$

Plug-in gradient estimator. Under standard regularity assumptions (Appendix B), differentiating the plug-in objective yields

$$\nabla_\theta J(\phi, \theta) = \mathbb{E}_{x\sim\mathcal{D}_x}\left[\left(\nabla_\theta w^\star_\theta(x)\right)^\top f_\phi(x)\right], \quad \nabla_\theta w^\star_\theta(x) \in \mathbb{R}^{d_w \times d_\theta}. \tag{4}$$

Since θ enters $J(\phi, \theta)$ solely through the sampling distribution, computing $\nabla_\theta J(\phi, \theta)$ reduces to differentiating $w^\star_\theta(x)$. For polyhedral $\mathcal{W}$ and other cases, the mapping $\hat{c} \mapsto w^\star(\hat{c})$ may be piecewise constant almost everywhere, so differentiating through it in the second expectation in (3) may yield uninformative gradients, making it difficult to obtain a one-sample stochastic gradient estimator of $\nabla_\theta J(\phi, \theta)$. One approach to overcome this challenge is to again rely on a score function representation:

$$g_t^{\text{plug-in}} = \left(w_t^\top f_{\phi_t}(x_t)\right) \nabla_\theta \log p_{\theta_t}(\hat{c}_t \mid x_t), \quad \text{where} \quad \hat{c}_t \sim p_{\theta_t}(\cdot \mid x_t), w_t = w^\star(\hat{c}_t). \tag{5}$$

Using the above estimator resembles a traditional actor-critic method; however, again it only weakly exploits the optimization structure.

Surrogate plug-in gradient estimator. To better exploit the linear optimization structure, we build on the extensive literature of decision-focused *surrogate* methods that have been developed in the full information setting. Namely, we replace $f_\phi(x)^\top w^\star(\hat{c})$ in (3) by a differentiable proxy $h^{\mathrm{sur}}(f_\phi(x), \hat{c})$ designed to provide informative gradients. Examples include the SPO+ loss (Elmachtoub & Grigas, 2022), the Perturbed Fenchel–Young loss (Berthet et al., 2020), and the Perturbation Gradient loss (Gupta & Huang, 2024), among others. For any choice of surrogate function $h^{\mathrm{sur}}(f_\phi(x), \hat{c})$, we define the surrogate plug-in objective

$$J^{\mathrm{sur}}(\phi, \theta) := \mathbb{E}_{x\sim\mathcal{D}_x, \hat{c}\sim p_\theta(\cdot|x)}\left[h^{\mathrm{sur}}(f_\phi(x),\, \hat{c})\right]. \tag{6}$$

Unlike the plug-in objective $J(\phi, \theta)$, which requires differentiating through $w^\star_\theta(x)$, we instead differentiate through the surrogate $h^{\mathrm{sur}}(f_\phi(x), \hat{c})$, which is engineered to have informative gradients almost everywhere. Combined with a reparameterization $\hat{c} = \mu_\theta(x, \varepsilon)$ with $\varepsilon \sim \nu$ independent of θ (Assumption 4; for the Gaussian policy, $\mu_\theta(x, \varepsilon) = \bar{\mu}_\theta(x) + \sigma_{\mathrm{G}}\varepsilon$ with $\nu = \mathcal{N}(0, I)$), this yields the surrogate gradient estimator $g_t^{\mathrm{sur}} = \big(\nabla_\theta \mu_{\theta_t}(x_t, \varepsilon_t)\big)^\top \nabla_{\hat{c}} h^{\mathrm{sur}}\big(f_{\phi_t}(x_t), \hat{c}_t\big)$. Conditioned on (ϕ_t, θ_t), g_t^{sur} is unbiased for $\nabla_\theta J^{\mathrm{sur}}(\phi_t, \theta_t)$; bias relative to V^{cost} comes from the nuisance estimate and the surrogate choice. The full surrogate framework is in Appendix E.

2.3 Decision-focused hybrid policy gradient algorithm

The score function gradient estimator (1) and the plug-in objective (3) (or its surrogate function variant (6)) provide two complementary gradient signals for θ. Our algorithm exploits both: at each iteration, it updates the nuisance parameter ϕ by least-squares SGD step on (2), and the parameter θ by an SGD step combining the two signals. This complementarity motivates combining them through a convex mixture. For a mixing parameter $\alpha \in [0, 1]$, we define the hybrid gradient estimator

$$g_t^\alpha = \alpha\, g_t^{\mathrm{score}} + (1 - \alpha)\, g_t^{\text{plug-in}} \tag{7}$$

The surrogate variant of the hybrid estimator is obtained by substituting g_t^{sur} for $g_t^{\text{plug-in}}$, yielding $g_t^{\alpha,\mathrm{sur}} := \alpha g_t^{\mathrm{score}} + (1 - \alpha) g_t^{\mathrm{sur}}$. At $\alpha = 1$ the estimator reduces to the unbiased score function gradient, while at $\alpha = 0$ it relies entirely on the plug-in gradient.

Algorithm 1 instantiates this construction as an online procedure covering both the plug-in and surrogate variants.

Algorithm 1: Decision-focused hybrid policy gradient (DFHPG)

Input: Initial parameters $\phi_1 \in \Phi$, $\theta_1 \in \mathbb{R}^{d_\theta}$; mixing parameter $\alpha \in [0, 1]$; step sizes $\eta_t, \beta > 0$
for $t = 1, 2, \ldots, T$ **do**
- Observe context x_t;
- Sample $\hat{c}_t \sim p_{\theta_t}(\cdot \mid x_t)$ and execute $w_t \in \arg\min_{w \in \mathcal{W}} \hat{c}_t^\top w$;
- Observe feedback v_t and set $y_t = e(w_t)^\top v_t$;
- $\phi_{t+1} = \Pi_\Phi\Big(\phi_t - \eta_t \nabla_\phi \ell_t(\phi_t)\Big), \quad \ell_t(\phi) = \|v_t - H(w_t) f_\phi(x_t)\|^2$ // **Nuisance update**
- $\theta_{t+1} = \theta_t - \beta\, \tilde{g}_t$, where $\tilde{g}_t = g_t^\alpha$ for the plug-in variant and $\tilde{g}_t = g_t^{\alpha,\mathrm{sur}}$ for the surrogate variant// **Hybrid policy update**

2.4 Convergence results

We formalize the convergence guarantees of Algorithm 1. Since the objective is non-convex in θ, we target standard non-convex stationarity criteria. The analysis hinges on a key structural property: the hybrid update is conditionally biased only through the nuisance error, which is in turn controlled by the regret of the nuisance update (Proposition 2.1). Under standard regularity formalized in Appendix D, the plug-in objective $J(\phi, \theta)$ is L_θ^J-smooth in θ with gradient norm bounded by G^J.

Theorem 2.2 (Non-convex stationarity, plug-in variant)**.** *Let* $V^\star := \inf_\theta V^{\mathrm{cost}}(\theta) > -\infty$, $\mathrm{Var}[g_t^{\mathrm{score}} \mid \phi_t, \theta_t] \le \sigma^2_{\mathrm{score}}$, *and* $\mathrm{Var}[g_t^{\text{plug-in}} \mid \phi_t, \theta_t] \le \sigma^2_{\text{plug-in}}$. *With step size* $\beta = 1/(L_\theta^J \sqrt{T})$, *Algorithm 1 satisfies*

(i) *With* $\eta_t = D_\Phi/(G_{\mathcal{R}}\sqrt{T})$*:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\big[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\big] \;\le\; \frac{2L_\theta^J \Delta_1 + (G^J)^2 + \sigma_\alpha^2}{\sqrt{T}} + \mathcal{O}\Big((1-\alpha)^2 \frac{D_\Phi G_{\mathcal{R}}}{\lambda\sqrt{T}}\Big), \tag{8}$$

(ii) *With* $\eta_t = 1/(\lambda t)$*:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\big[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\big] \;\le\; \frac{2L_\theta^J \Delta_1 + (G^J)^2 + \sigma_\alpha^2}{\sqrt{T}} + \mathcal{O}\Big(\frac{(1-\alpha)^2(1+\log T)}{\lambda^2 T}\Big), \tag{9}$$

where $\Delta_1 := V^{\mathrm{cost}}(\theta_1) - V^\star$*, and the variance* $\sigma_\alpha^2 := (\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\text{plug-in}})^2$.

In both cases, the optimization term is $\mathcal{O}(T^{-1/2})$, matching the standard non-convex SGD rate (Ghadimi & Lan, 2013). The constant stepsize in (i) yields a $\mathcal{O}(T^{-1/2})$ bias term, while the adaptive schedule in (ii) exploits strong convexity to sharpen the bias to $\mathcal{O}(\log T/T)$. Supporting lemmas and proofs are deferred to Appendices C–E. The variance σ_α^2 interpolates continuously between two extremes. At $\alpha = 1$ one recovers the unbiased score function gradient (Sutton & Barto, 2018). At $\alpha = 0$ one relies entirely on the plug-in gradient, which enjoys lower variance once f_{ϕ_t} is well-trained, at the cost of a nuisance-driven bias. The optimal α shifts toward zero as the nuisance estimation improves, suggesting that adaptive schedules $\alpha_t \to 0$ may improve empirical performance.

We now turn to the surrogate variant. Define $V^\alpha(\theta) := \alpha\, V^{\mathrm{cost}}(\theta) + (1-\alpha)\, J^{\mathrm{sur}}(\phi^\star, \theta)$. Under standard regularity formalized in Appendix E, V^α is L_θ^α-smooth in θ with gradient norm bounded by a constant $\tilde{G}_\alpha$ (Appendix E). We state the result under the dynamic schedule $\eta_t = 1/(\lambda t)$; an analogous bound to Theorem 2.2(i) holds for the constant step size for our surrogate model.

Theorem 2.3 (Non-convex stationarity, surrogate variant)**.** *Let* $V^{\alpha,\star} := \inf_\theta V^\alpha(\theta)$. *Running Algorithm 1 with* $\beta = 1/(L_\theta^\alpha\sqrt{T})$ *and* $\eta_t = 1/(\lambda t)$ *yields*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla V^\alpha(\theta_t)\|^2\big] \le \frac{2L_\theta^\alpha\Delta_1^\alpha + \tilde{G}_\alpha^2 + \tilde{\sigma}_\alpha^2}{\sqrt{T}} + \mathcal{O}\Big((1-\alpha)^2\,\tfrac{1+\log T}{\lambda^2 T}\Big). \tag{10}$$

with $\Delta_1^\alpha = V^\alpha(\theta_1) - V^{\alpha,\star}$, $\tilde{\sigma}_\alpha^2 := (\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\mathrm{sur}})^2$, $\sigma_{\mathrm{sur}}^2 \ge \mathrm{Var}[g_t^{\mathrm{sur}} \mid \phi_t, \theta_t]$.

Theorem 2.3 targets the surrogate-induced objective V^α. How closely V^α tracks V^{cost} depends on the surrogate. For SPO+ and PG, calibration results relate surrogate risk to decision regret in the offline, full-information setting (Liu & Grigas, 2021; Gupta & Huang, 2024).

Scope of the analysis. Theorems 2.2 and 2.3 analyze the fixed-α update with a constant policy step size, the standard setting for non-convex stochastic gradient methods (Ghadimi & Lan, 2013), and a constant or decreasing nuisance step size. They allow any variance-reduction baseline independent of $\hat{c}_t$ given x_t. For the plug-in term, Theorem 2.3 covers surrogates with bounded, Lipschitz gradients (Assumption 9). The experiments use standard DFL surrogates (SPO+, PG, Differentiable Black-Box (DBB), and ranking losses), which are built from oracle solutions and are therefore non-smooth in $\hat{c}$, so the evidence for them is empirical. The experiments add four practical adaptations that the theorems do not model: a mixing weight α_t that decays over iterations, a step-size schedule, separate L_2 normalization of the score function and plug-in gradients before they are mixed, and a neural-network nuisance model.

3 Numerical experiments

3.1 Benchmark problems

The experiments use four online contextual optimization benchmarks with bandit feedback: three synthetic linear-objective tasks (top-k selection, shortest path, and combinatorial pricing) and a real-data energy-scheduling instance. Benchmark formulations and data details (synthetic data generation and the energy-scheduling data source and chronological split) are collected in Appendix F; semi-bandit and full-information variants of the feedback model are reported in Appendix J.7.

Benchmarks. *Top-k selection* chooses k of $d = 15$ items, and the item costs are a polynomial of a 5-dimensional context with multiplicative noise, whose degree sets the nonlinearity. *Shortest path* routes one unit of flow across a 5×5 grid with 40 edges, and the edge costs follow the same construction from a 10-dimensional context. *Combinatorial pricing* assigns one of 4 price levels to each of 20 products under a promotion budget. The uncertainty is the demand for each product at each price given a 25-dimensional context, and the cost is the negative realized revenue. *Energy scheduling* assigns tasks to machines and start slots under resource capacities. The uncertainty is the electricity price in each of the 48 half-hour slots of a day, predicted from 8 features per slot in the

empirical Single Electricity Market Operator (SEMO) data of Mandi et al. (2024), and each iteration is one day.

3.2 Experimental setup

The proposed method is reported as DFHPG. It uses the hybrid gradient update in Algorithm 1, with the fixed mixing weight α replaced by a time-varying α_t decaying from $\alpha_{\max}$ to $\alpha_{\min}$. The score function gradient subtracts a variance-reduction baseline b_t. Two ablations isolate the two gradient components: DFHPG-1 uses only the score function update, DFHPG-0 only the plug-in update. Appendix G specifies these components, and Appendix I the tuning and selected configurations.

We compare against three contextual bandit (CB) baselines that share DFHPG's context, feedback, and linear optimization oracle, isolating the effect of the learning update from the action-selection rule. GreedyCB uses greedy exploitation with a point cost predictor; ϵ-GreedyCB adds random exploration; TSCB samples a cost vector from the learned conditional cost distribution and solves the linear optimization at that sample, matching Thompson sampling's action-level randomization without an explicit Bayesian posterior. We also compare with two standard linear bandit algorithms. Optimism in the face of uncertainty for linear bandits (OFUL) (Abbasi-Yadkori et al., 2011) has a confidence bonus that is nonlinear in the decision, so our implementation enumerates the feasible set, and we run it on top-k and shortest path. Linear Thompson sampling (LinTS) (Agrawal & Goyal, 2013) keeps the linear oracle and maintains a closed-form Gaussian posterior over the cost-model parameter, so we run it on all four benchmarks with each benchmark's cost model. Pseudocode and settings for all baselines are in Appendix H.

The default distributional cost-vector prediction model is the Gaussian linear model; the Gaussian neural-network (NN), conditional normalizing-flow (CNF), and denoising diffusion probabilistic model (DDPM) variants are also supported and reported as alternatives in Appendix J.3. The Gaussian variants set $p_\theta(\cdot \mid x) = \mathcal{N}(\bar{\mu}_\theta(x),\, \sigma_{\mathrm{G}}^2 I_{d_w})$, where $\bar{\mu}_\theta$ is the mean prediction function parameterized either as a linear map (Gaussian linear) or as a two-layer neural network with 128 hidden units (Gaussian NN), and $\sigma_{\mathrm{G}} > 0$ is a fixed exploration scale. Update rules for the generative variants are in Appendix G.4. The main metric reported is cumulative regret, $\mathrm{Regret}_T = \sum_{t=1}^{T} \big(f^\star(x_t)^\top w_t - \min_{w \in \mathcal{W}} f^\star(x_t)^\top w\big)$, where $w_t \in \mathcal{W}$ is the decision implemented at iteration t and $f^\star(x_t) = \mathbb{E}[c_t \mid x_t]$ is the mean cost. Convergence figures plot the ergodic average regret per iteration, $\mathrm{Regret}_t\,/t$.

3.3 Results

Table 1 reports final cumulative regret across the four benchmarks under both feedback regimes. Under pure bandit feedback, DFHPG achieves the lowest mean regret on top-k selection and pricing, and the lowest among the contextual-bandit methods on shortest path; on energy, both DFHPG and DFHPG-0 beat the contextual-bandit baselines by roughly 1.7–1.9×, and the two perform alike (each lower in about half of the 30 replications), while DFHPG-1 has far higher regret. DFHPG-0 is the $\alpha = 0$ member of the DFHPG family, and the score term helps on top-k and shortest path but adds little on pricing and energy (Appendix J.2). Semi-bandit feedback lowers the cumulative regret of DFHPG by roughly 26–58% on the synthetic benchmarks and ∼39% on energy, and DFHPG-0 has lower mean regret than DFHPG in every column: the coordinate-level signal feeds the nuisance fit directly, so the score function term is most useful in the pure-bandit regime. On pricing, the semi-bandit versions of the contextual-bandit baselines are lower still (Appendix J.7). DFHPG has lower mean regret than OFUL and LinTS on top-k and lower regret than LinTS in 27 of 30 energy replications. On shortest path, DFHPG stays competitive with OFUL and LinTS but does not beat them: it has higher regret than OFUL in 19 and than LinTS in 16 of the 30 replications. Here the linear bandits benefit from re-solving the regularized least-squares problem for the linear cost model after every round, whereas DFHPG takes one stochastic gradient step, and with only 70 feasible paths OFUL can also score every path. On pricing, DFHPG has 4% lower regret than LinTS. More broadly, OFUL and LinTS need a cost model that is linear in its parameters, which keeps their least-squares updates and posterior in closed form, whereas DFHPG also runs with the neural, flow, and diffusion cost models of Appendix J.3.

Stationarity. Theorems 2.2 and 2.3 guarantee stationarity of V^{cost} and V^{α}, whereas Table 1 measures regret, and the two connect through $\nabla_\theta V^{\mathrm{cost}}$, the gradient of expected per-round regret.

Table 1: Final cumulative regret (mean $\pm$ standard error, 30 replications) with the Gaussian linear cost model: top-k and shortest path at degree 8 and $T = 2{,}000$, pricing at degree 6 and $T = 10{,}000$. Best in bold, second-best underlined per feedback block. OFUL runs where the feasible set can be enumerated.

Method	Top-k (deg. 8)	Shortest path (deg. 8)	Pricing (deg. 6)	Energy
Pure bandit feedback				
GREEDYCB	$6.94{\times}10^5 \pm 3.38{\times}10^4$	$1.98{\times}10^6 \pm 4.40{\times}10^4$	$7.38{\times}10^4 \pm 1.6{\times}10^2$	$6.43{\times}10^7 \pm 1.95{\times}10^5$
ϵ-GREEDYCB	$7.17{\times}10^5 \pm 3.98{\times}10^4$	$2.06{\times}10^6 \pm 4.12{\times}10^4$	$7.61{\times}10^4 \pm 1.8{\times}10^2$	$6.58{\times}10^7 \pm 3.01{\times}10^5$
TSCB	$6.93{\times}10^5 \pm 3.87{\times}10^4$	$2.00{\times}10^6 \pm 4.37{\times}10^4$	$7.38{\times}10^4 \pm 2.0{\times}10^2$	$6.43{\times}10^7 \pm 1.94{\times}10^5$
OFUL	$1.70{\times}10^5 \pm 1.45{\times}10^4$	$\mathbf{1.42{\times}10^6 \pm 3.73{\times}10^4}$	–	–
LinTS	$1.79{\times}10^5 \pm 1.90{\times}10^4$	$1.45{\times}10^6 \pm 3.66{\times}10^4$	$7.29{\times}10^4 \pm 2.4{\times}10^2$	$5.50{\times}10^7 \pm 1.99{\times}10^5$
DFHPG-1	$4.00{\times}10^5 \pm 4.49{\times}10^4$	$1.81{\times}10^6 \pm 6.08{\times}10^4$	$8.55{\times}10^4 \pm 2.7{\times}10^2$	$1.60{\times}10^8 \pm 1.86{\times}10^7$
DFHPG	$\mathbf{1.68{\times}10^5 \pm 1.47{\times}10^4}$	$1.50{\times}10^6 \pm 5.04{\times}10^4$	$\mathbf{7.00{\times}10^4 \pm 2.9{\times}10^2}$	$3.85{\times}10^7 \pm 8.79{\times}10^6$
DFHPG-0	$1.99{\times}10^5 \pm 2.38{\times}10^4$	$1.58{\times}10^6 \pm 5.15{\times}10^4$	$7.02{\times}10^4 \pm 3.6{\times}10^2$	$\mathbf{3.50{\times}10^7 \pm 8.12{\times}10^6}$
Semi-bandit feedback				
DFHPG	$7.10{\times}10^4 \pm 8.52{\times}10^3$	$6.83{\times}10^5 \pm 3.11{\times}10^4$	$5.19{\times}10^4 \pm 1.0{\times}10^2$	$2.35{\times}10^7 \pm 4.21{\times}10^5$
DFHPG-0	$\mathbf{6.65{\times}10^4 \pm 5.30{\times}10^3}$	$\mathbf{5.31{\times}10^5 \pm 2.28{\times}10^4}$	$\mathbf{5.18{\times}10^4 \pm 9.8{\times}10^1}$	$\mathbf{2.33{\times}10^7 \pm 3.78{\times}10^5}$

Because $f^\star$ is known on the synthetic benchmarks, we estimate $\|\nabla_\theta V^{\text{cost}}(\theta_t)\|^2$ along the fixed-α runs at $\alpha = 0$ (constant step sizes, no gradient normalization). Its ergodic average falls $4.3\times$ on top-k and $8.5\times$ on shortest path between $t = 100$ and 2000, together with per-round regret (Appendix J.2).

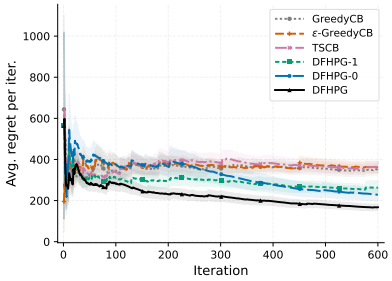


Figure 1: Top-k, first 600 iterations.

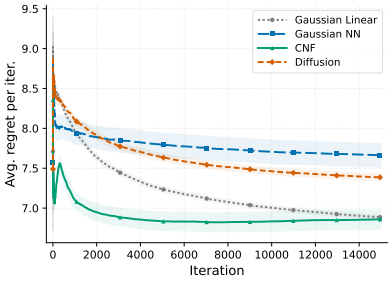


Figure 2: Cost models on pricing.

Figure 1 traces top-k dynamics over the first 600 iterations and shows that the score function and plug-in components are complementary in time. For the first $\sim$300 iterations DFHPG-1 sits below DFHPG-0: the score function gradient is unbiased and immediately usable, while the plug-in gradient relies on a nuisance that has not accumulated enough data. The order reverses past $\sim$300 iterations, when the nuisance stabilizes and the lower-variance plug-in signal pulls DFHPG-0 below DFHPG-1; through its time-varying weighting, hybrid DFHPG follows the better component in each phase and ends below the contextual-bandit baselines.

Figure 2 compares distributional cost models on pricing at $T = 15{,}000$: CNF has the lowest regret for most of the run but levels off after about 5,000 iterations, while the Gaussian linear model keeps improving and nearly matches it by the end (Appendix J.3).

4 CONCLUSION

DFHPG couples an unbiased score function update with a decision-focused plug-in update, and under the stated assumptions its fixed-weight update reaches first-order stationarity at rate $\mathcal{O}(T^{-1/2})$. Under bandit feedback, DFHPG and the plug-in-only DFHPG-0 reduce cumulative regret relative to the contextual-bandit baselines and the score-only ablation on all four benchmarks, with DFHPG strongest on the synthetic benchmarks and DFHPG-0 on energy. DFHPG is also more flexible than linear bandit algorithms: it needs neither a closed-form posterior (LinTS) nor an enumerable feasible set (OFUL), only a distributional cost model with the required gradient estimators and a linear optimization oracle. Extending the framework beyond linear feedback and risk-neutral objectives is a natural next step.

Reproducibility statement

Anonymized code that reproduces every experiment, including the benchmark generators, all methods, and the tuning and plotting scripts, is provided in the supplementary material. Appendix F specifies the four benchmarks, including the data source and chronological split for energy scheduling. Appendices G and H give the implementation details of DFHPG and pseudocode and settings for every baseline. Appendix I describes the tuning protocol, the selected hyperparameters, the number of replications, and the compute used. Appendices A–E state all assumptions and contain the proofs.

AI use statement

We used Claude Code to polish the writing of this paper. The authors reviewed all AI-assisted edits. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 263–274, 2008.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 127–135, 2013.

Omar Bennouna, Amine Bennouna, Saurabh Amin, and Asuman Ozdaglar. What data enables optimal decisions? An exact characterization for linear optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 184042–184071, 2025. doi: 10.52202/085713-5534.

Omar Bennouna, Amine Bennouna, Saurabh Amin, and Asuman Ozdaglar. Data informativeness in linear optimization under uncertainty, 2026. URL https://arxiv.org/abs/2602.15365.

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, pp. 9508–9519, 2020.

Louis Bouvier, Thibault Prunet, Vincent Leclère, and Axel Parmentier. Primal-dual algorithm for contextual stochastic combinatorial optimization, 2025.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. doi: 10.1561/2200000024. URL https://doi.org/10.1561/2200000024.

Aymeric Capitaine, Maxime Haddouche, Eric Moulines, Michael I. Jordan, Etienne Boursier, and Alain Oliviero Durmus. Online decision-focused learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FJhtHBphCt.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 355–366, 2008.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

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

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pp. 385–394, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Vishal Gupta and Michael Huang. Decision-focused learning with directional gradients. *Advances in Neural Information Processing Systems*, 37:79194–79220, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Xinyi Hu, Jasper C. H. Lee, Jimmy H. M. Lee, and Peter J. Stuckey. Multi-stage predict+optimize for (mixed integer) linear programs. *Advances in Neural Information Processing Systems*, 37:64794–64827, 2024a. doi: 10.52202/079017-2068. URL https://openreview.net/forum?id=pXFiHHySEw.

Yichun Hu, Nathan Kallus, Xiaojie Mao, and Yanchen Wu. Contextual linear optimization with bandit feedback. *Advances in Neural Information Processing Systems*, 37:26976–27004, 2024b.

Yichun Hu, Nathan Kallus, Xiaojie Mao, and Yanchen Wu. Contextual linear optimization under partial feedback. *arXiv preprint arXiv:2405.16564*, 2025.

Shinji Ito, Daisuke Hatano, Hanna Sumita, Kei Takemura, Takuro Fukunaga, Naonori Kakimura, and Ken-Ichi Kawarabayashi. Oracle-efficient algorithms for online linear optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1008–1014, 2000.

Julia Kreutzer, Artem Sokolov, and Stefan Riezler. Bandit structured prediction for neural sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1503–1513, 2017.

Akshay Krishnamurthy, Alekh Agarwal, and Miroslav Dudik. Contextual semibandits via supervised learning oracles. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Heyuan Liu and Paul Grigas. Risk bounds and calibration for a smart predict-then-optimize method. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22083–22094, 2021.

Heyuan Liu and Paul Grigas. Online contextual decision-making with a smart predict-then-optimize method, 2022. URL https://arxiv.org/abs/2206.07316.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14935–14947, 2022.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor 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. URL https://doi.org/10.1613/jair.1.15320.

Pasquale Minervini, Luca Franceschi, and Mathias Niepert. Adaptive perturbation-based gradient estimation for discrete latent variable models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9200–9208, 2023. doi: 10.1609/aaai.v37i8.26103.

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.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit MLE: Backpropagating through discrete exponential family distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14567–14579, 2021.

Francesco Orabona. A modern introduction to online learning, 2019. URL https://arxiv.org/abs/1912.13213.

Warren B Powell. Designing lookahead policies for sequential decision problems in transportation and logistics. *IEEE Open Journal of Intelligent Transportation Systems*, 3:313–327, 2022.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025.

Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through combinatorial algorithms: Identity with projection works. In *International Conference on Learning Representations (ICLR)*, 2023.

Mattia Silvestri, Senne Berden, Gaetano Signorelli, Ali İrfan Mahmutoğulları, Jayanta Mandi, Brandon Amos, Tias Guns, and Michele Lombardi. Score function gradient estimation to widen the applicability of decision-focused learning. *Journal of Artificial Intelligence Research*, 85(10): 10:1–10:34, 2026. doi: 10.1613/jair.1.19498. URL https://doi.org/10.1613/jair.1.19498.

Artem Sokolov, Julia Kreutzer, Stefan Riezler, and Christopher Lo. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1057–1063, 2000.

Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 16(3): 297–335, 2024. doi: 10.1007/s12532-024-00255-x. URL https://doi.org/10.1007/s12532-024-00255-x.

Pascal Van Hentenryck and Russell Bent. *Online stochastic combinatorial optimization*. The MIT Press, 2006.

Marin Vlastelica Pogančić, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations (ICLR)*, 2020.

Prince Zizhuang Wang, Shuyi Chen, Jinhao Liang, Ferdinando Fioretto, and Shixiang Zhu. Gen-DFL: Decision-focused generative learning for robust decision making. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GU2197a3Lm.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Tinghan Ye, Sikai Cheng, Amira Hijazi, and Pascal Van Hentenryck. Contextual stochastic optimization for omnichannel multicourier order fulfillment under delivery time uncertainty. *Manufacturing & Service Operations Management*, 28(4):1068–1090, 2026a. doi: 10.1287/msom.2024.1328. URL https://doi.org/10.1287/msom.2024.1328.

Tinghan Ye, Shuaicheng Tong, Changkun Guan, Beste Basciftci, and Pascal Van Hentenryck. Learning optimization proxies for sequential contextual stochastic programs: An order fulfillment application, 2026b. URL https://arxiv.org/abs/2606.25362.

Zihao Zhao, Christopher Yeh, Lingkai Kong, and Kai Wang. Diffusion-DFL: Decision-focused diffusion models for stochastic optimization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uhv3f80jmG.

Yinglun Zhu, Dylan J. Foster, John Langford, and Paul Mineiro. Contextual bandits with large action spaces: Made practical. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27428–27453, 2022.

Lukas Zierahn, Dirk van der Hoeven, Nicolò Cesa-Bianchi, and Gergely Neu. Nonstochastic contextual combinatorial bandits. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8771–8813, 2023.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

Organization. Appendices A–E contain the feedback examples and the proofs. Appendices F–I give the experimental details: benchmarks, DFHPG implementation, baselines, and hyperparameters. Appendix J reports additional results.

A Partial feedback structure examples

The feedback structure in Assumption 1 is flexible enough to capture several practically important settings. We highlight the three main instances considered in this paper.

- **Full feedback.** Set $H(w) = I$ and $e(w) = w$. The feedback coincides with the cost vector, $v = c$, and the linkage identity $c^\top w = e(w)^\top v$ holds trivially.
- **Bandit feedback.** Set $H(w) = w^\top$ and $e(w) = 1$. Then $v = c^\top w$ is the scalar realized cost, and the linkage identity reduces to $c^\top w = e(w)^\top v$, which holds trivially.
- **Semi-bandit / combinatorial feedback.** Let $\mathcal{W} \subseteq \{0,1\}^{d_w}$. Set $H(w) = \mathrm{diag}(w)$ and $e(w) = \mathbf{1}$. Then the feedback $v = \mathrm{diag}(w)c$ records the cost of each selected coordinate. The linkage condition holds since $H(w)^\top e(w) = \mathrm{diag}(w)\mathbf{1} = w$, and the identity $c^\top w = e(w)^\top v$ follows immediately.

B Regularity of the perturbed optimizer and score function representation

We consider the perturbed optimizers $w^\star_\theta(x) := \mathbb{E}_{\hat{c}\sim p_\theta(\cdot|x)}\big[w^\star(\hat{c})\big]$. The smoothing induced by p_θ makes $w^\star_\theta(x)$ differentiable in θ even though the underlying optimizer $\hat{c} \mapsto w^\star(\hat{c})$ is piecewise constant. The following assumptions and supporting results formalize this differentiability and the resulting regularity of the policy value V^{cost}.

Assumption 2 (Bounded feasible set and costs)**.** *$\mathcal{W}$ is compact with $B_{\mathcal{W}} := \sup_{w\in\mathcal{W}} \|w\| < \infty$, and there exists $B_c > 0$ such that $\|c_t\| \le B_c$ almost surely.*

Assumption 3 (Smooth parametric family)**.** *The parametric family $p_\theta(\cdot \mid x)$ is twice continuously differentiable in θ for all $x \in \mathcal{X}$, and there exist constants $M, M_\nabla < \infty$ such that for all $\theta, \theta' \in \mathbb{R}^{d_\theta}$ and $x \in \mathcal{X}$,*

$$\int \|\nabla_\theta p_\theta(\hat{c} \mid x)\| \,\mathrm{d}\hat{c} \le M \tag{11}$$

$$\int \|\nabla_\theta p_\theta(\hat{c} \mid x) - \nabla_\theta p_{\theta'}(\hat{c} \mid x)\| \,\mathrm{d}\hat{c} \le M_\nabla \|\theta - \theta'\| \tag{12}$$

Conditions (11)–(12) bound the parametric family in L^1 on the gradient and its variation, and are satisfied by all common smooth parameterizations, e.g., Gaussian families with smoothly parameterized mean and covariance, or exponential families with bounded sufficient statistics. They ensure that $\theta \mapsto \nabla_\theta w^\star_\theta(x)$ is itself Lipschitz, which is needed to establish L^J_θ-smoothness of V^{cost} in Lemma D.1(ii).

Assumption 4 (Reparameterization)**.** *Under $p_\theta(\cdot \mid x)$, we assume the reparameterization $\hat{c} = \mu_\theta(x, \varepsilon)$ with $\varepsilon \sim \nu$ independent of θ. The map μ_θ is continuously differentiable in θ with $\|\nabla_\theta \mu_\theta(x, \varepsilon)\| \le G_\mu$ almost surely for all $\theta \in \mathbb{R}^{d_\theta}$ and $x \in \mathcal{X}$, and there exists $L_\mu < \infty$ such that $\|\nabla_\theta \mu_\theta(x, \varepsilon) - \nabla_\theta \mu_{\theta'}(x, \varepsilon)\| \le L_\mu \|\theta - \theta'\|$ almost surely.*

Corollary B.1 (Bounded and Lipschitz mean-decision gradient)**.** *Under Assumptions 2 and 3, $\theta \mapsto w^\star_\theta(x)$ is differentiable for every $x \in \mathcal{X}$, with the score function representation*

$$\nabla_\theta w^\star_\theta(x) = \mathbb{E}_{\hat{c} \sim p_\theta(\cdot|x)}\big[w^\star(\hat{c})\, \nabla_\theta \log p_\theta(\hat{c} \mid x)^\top\big]. \tag{13}$$

Moreover, the Jacobian satisfies the bound

$$\|\nabla_\theta w^\star_\theta(x)\| \;\le\; B_{\nabla w^\star_\theta} := B_{\mathcal{W}}\, M, \qquad \forall \theta \in \mathbb{R}^{d_\theta},\ x \in \mathcal{X}, \tag{14}$$

and is L_w-Lipschitz in θ with $L_w := B_{\mathcal{W}}\, M_\nabla$:

$$\|\nabla_\theta w^\star_\theta(x) - \nabla_\theta w^\star_{\theta'}(x)\| \;\le\; L_w\, \|\theta - \theta'\|, \qquad \forall \theta, \theta' \in \mathbb{R}^{d_\theta},\ x \in \mathcal{X}. \tag{15}$$

Proof. Under Assumption 3, dominated convergence applied to $w^\star_\theta(x) = \int w^\star(\hat{c})\, p_\theta(\hat{c} \mid x)\, d\hat{c}$ yields

$$\nabla_\theta w^\star_\theta(x) = \int w^\star(\hat{c})\, \nabla_\theta p_\theta(\hat{c} \mid x)\, d\hat{c} = \mathbb{E}_{\hat{c} \sim p_\theta(\cdot|x)}\big[w^\star(\hat{c})\, \nabla_\theta \log p_\theta(\hat{c} \mid x)^\top\big],$$

where the second equality uses the log-derivative identity $\nabla_\theta p_\theta = p_\theta\, \nabla_\theta \log p_\theta$.

Boundedness. Applying the triangle inequality to (13) in integral form, together with $\|w^\star(\hat{c})\| \le B_{\mathcal{W}}$ (Assumption 2) and (11),

$$\|\nabla_\theta w^\star_\theta(x)\| \;\le\; \int \|w^\star(\hat{c})\|\, \|\nabla_\theta p_\theta(\hat{c} \mid x)\|\, \mathrm{d}\hat{c} \;\le\; B_{\mathcal{W}} \int \|\nabla_\theta p_\theta(\hat{c} \mid x)\|\, \mathrm{d}\hat{c} \;\le\; B_{\mathcal{W}}\, M,$$

Lipschitz continuity. Subtracting the integral representations at θ and θ',

$$\nabla_\theta w^\star_\theta(x) - \nabla_\theta w^\star_{\theta'}(x) = \int w^\star(\hat{c})\, \big(\nabla_\theta p_\theta(\hat{c} \mid x) - \nabla_\theta p_{\theta'}(\hat{c} \mid x)\big)^\top\, \mathrm{d}\hat{c}.$$

Applying the triangle inequality, $\|w^\star(\hat{c})\| \le B_{\mathcal{W}}$, and (12) yields

$$\|\nabla_\theta w^\star_\theta(x) - \nabla_\theta w^\star_{\theta'}(x)\| \;\le\; B_{\mathcal{W}} \int \|\nabla_\theta p_\theta(\hat{c} \mid x) - \nabla_\theta p_{\theta'}(\hat{c} \mid x)\|\, \mathrm{d}\hat{c} \;\le\; B_{\mathcal{W}}\, M_\nabla\, \|\theta - \theta'\|.$$

Setting $L_w := B_{\mathcal{W}}\, M_\nabla$ yields (15). □

C Online regret analysis of the nuisance estimator

The nuisance update can be analyzed as an online convex optimization problem independently of the policy sequence $\{\theta_t\}_{t=1}^T$. Rather than tracking the moving per-iteration minimizer $\arg\min_{\phi \in \Phi} \mathcal{R}_{\theta_t}(f_\phi)$, we compare against the fixed realizer $\phi^\star \in \Phi$ of Assumption 5 below, a valid comparison since $f^\star$ is the universal minimizer of $\mathcal{R}_\theta$ for every θ simultaneously (Remark 2.1). To analyze the nuisance update via online convex optimization arguments, we make the following assumptions, standard in the OCO literature (Orabona, 2019).

Assumption 5 (Realizability)**.** *There exists $\phi^\star \in \Phi$ such that $f_{\phi^\star} = f^\star$ (so $f^\star \in \mathcal{F} := \{f_\phi : \phi \in \Phi\}$ with an explicit parameter-space comparator).*

Assumption 6. *(i) Φ is convex and compact with diameter $D_\Phi := \sup_{\phi, \phi' \in \Phi} \|\phi - \phi'\| < \infty$.*

(ii) There exists $G_{\mathcal{R}} < \infty$ such that $\|\nabla_\phi \ell_t(\phi)\| \le G_{\mathcal{R}}$ almost surely for all t and $\phi \in \Phi$.

(iii) There exists $\lambda > 0$ such that, for every $\theta \in \mathbb{R}^{d_\theta}$, $\nabla^2_\phi \mathcal{R}_\theta(f_\phi) \;\succeq\; \lambda\, I_{d_\phi}$.

Assumption 6(i)–(ii) are standard online convex optimization assumptions, bounding the feasible set diameter and gradient magnitudes respectively. Assumption 6(iii) requires the nuisance objective to be strongly convex in ϕ, uniformly over θ; We first restate Proposition 2.1 for an arbitrary comparator $u \in \Phi$ and give its proof; the main-text statement is the case $u = \phi^\star$, for which $f_u = f^\star$.

Proposition C.1 (Nuisance regret)**.** *Under Assumptions 5–6, for any sequence $\{\theta_t\}_{t=1}^T$ and any fixed comparator $u \in \Phi$:*

(i) *With $\eta_t = D_\Phi/(G_\mathcal{R}\sqrt{T})$*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \;\le\; \frac{D_\Phi G_\mathcal{R}}{\sqrt{T}}$$

(ii) *With $\eta_t = 1/(\lambda t)$*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \;\le\; \frac{G_\mathcal{R}^2(1+\log T)}{2\lambda T}$$

Proof. Both bounds follow from standard online convex optimization arguments (Orabona, 2019); we include the derivations for completeness. By non-expansiveness of the Euclidean projection Π_Φ and the update rule $\phi_{t+1} = \Pi_\Phi(\phi_t - \eta_t g_t)$, where $g_t := \nabla_\phi \ell_t(\phi_t)$ is the stochastic gradient in Algorithm 1,

$$\|\phi_{t+1} - u\|^2 \le \|\phi_t - \eta_t g_t - u\|^2 = \|\phi_t - u\|^2 - 2\eta_t\langle g_t, \phi_t - u\rangle + \eta_t^2\|g_t\|^2 \tag{16}$$

Taking expectations, using unbiasedness of g_t and convexity of $\mathcal{R}_{\theta_t}$ in ϕ,

$$\mathbb{E}\big[\langle \nabla_\phi \mathcal{R}_{\theta_t}(f_{\phi_t}), \phi_t - u\rangle\big] \ge \mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big]$$

Substituting into (16) and rearranging,

$$2\eta_t\,\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \le \mathbb{E}\big[\|\phi_t - u\|^2\big] - \mathbb{E}\big[\|\phi_{t+1} - u\|^2\big] + \eta_t^2 G_\mathcal{R}^2$$

Summing over $t = 1, \dots, T$, using a constant step-size $\eta_t = \eta$ and telescoping,

$$2\eta\sum_{t=1}^{T}\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \le D_\Phi^2 + \eta^2 G_\mathcal{R}^2 T$$

Setting $\eta = D_\Phi/(G_\mathcal{R}\sqrt{T})$ yields the claimed bound. The bound (ii) follows from standard strong-convexity arguments in online convex optimization. When $\mathcal{R}_{\theta_t}$ is λ-strongly convex in ϕ,

$$\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \le \mathbb{E}\big[\langle g_t, \phi_t - u\rangle\big] - \frac{\lambda}{2}\mathbb{E}\big[\|\phi_t - u\|^2\big] \tag{17}$$

Non-expansiveness of Π_Φ gives $\mathbb{E}[\langle g_t, \phi_t - u\rangle] \le \frac{1}{2\eta_t}(\mathbb{E}[\|\phi_t - u\|^2] - \mathbb{E}[\|\phi_{t+1} - u\|^2]) + \frac{\eta_t G_\mathcal{R}^2}{2}$ Substituting into (17) and choosing $\eta_t = 1/(\lambda t)$,

$$\mathbb{E}\big[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)\big] \le \frac{\lambda(t-1)}{2}\mathbb{E}\big[\|\phi_t - u\|^2\big] - \frac{\lambda t}{2}\mathbb{E}\big[\|\phi_{t+1} - u\|^2\big] + \frac{G_\mathcal{R}^2}{2\lambda t}$$

Summing telescopes the distance terms (with non-positive boundary contribution), giving $\sum_{t=1}^{T}\mathbb{E}[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f_u)] \le \frac{G_\mathcal{R}^2(1+\log T)}{2\lambda}$ Dividing by T completes the proof. □

D Convergence analysis: plug-in hybrid variant

The nuisance regret translates into a bound on the gradient bias by combining strong convexity of $\mathcal{R}_\theta(\cdot)$ with the Lipschitz parameterization of $f_\phi(\cdot)$. We define the plug-in bias as

$$B(\phi, \theta) := J(\phi, \theta) - V^{\mathrm{cost}}(\theta).$$

Under realizability (Assumption 5)

$$V^{\mathrm{cost}}(\theta) = J(\phi^\star, \theta)$$

so equivalently

$$B(\phi, \theta) = J(\phi, \theta) - J(\phi^\star, \theta)$$

Assumption 7. *The nuisance estimator function class is bounded and L_{f_ϕ}-Lipschitz, i.e., there exists $B_{f_\phi} < \infty$ such that $\|f_\phi(x)\| \le B_{f_\phi}$ and $\|f_\phi(x) - f_{\phi'}(x)\| \le L_{f_\phi} \|\phi - \phi'\|$ for all $\phi \in \Phi$ and $x \in \mathcal{X}$.*

Lemma D.1 (Regularity of J)**.** *Under Assumptions 2,7 and Corollary B.1, the plug-in policy objective $J(\phi, \theta)$ satisfies:*

(i) $\|\nabla_\theta J(\phi, \theta)\| \le G^J := B_{f_\phi} B_{\nabla w^\star_\theta}$;

(ii) $\|\nabla_\theta J(\phi, \theta) - \nabla_\theta J(\phi, \theta')\| \le L^J_\theta \|\theta - \theta'\|$ *with* $L^J_\theta := B_{f_\phi} L_w = B_{f_\phi} B_{\mathcal{W}} M_\nabla$;

(iii) $|J(\phi, \theta) - J(\phi', \theta)| \le L^J_\phi \|\phi - \phi'\|$ *with* $L^J_\phi := B_{\mathcal{W}} L_{f_\phi}$.

Proof. Part (i). Differentiating under the expectation, $\nabla_\theta J(\phi, \theta) = \mathbb{E}_x[(\nabla_\theta w^\star_\theta(x))^\top f_\phi(x)]$, so $\|\nabla_\theta J(\phi, \theta)\| \le \mathbb{E}_x[\|\nabla_\theta w^\star_\theta(x)\| \|f_\phi(x)\|] \le B_{\nabla w^\star_\theta} B_{f_\phi}$, where the last step uses Corollary B.1 and Assumption 7.

Part (ii). For any $\theta, \theta' \in \mathbb{R}^{d_\theta}$,

$$\begin{aligned}\|\nabla_\theta J(\phi, \theta) - \nabla_\theta J(\phi, \theta')\| &\le \mathbb{E}_x\big[\|f_\phi(x)\| \|\nabla_\theta w^\star_\theta(x) - \nabla_\theta w^\star_{\theta'}(x)\|\big] \\ &\le B_{f_\phi} L_w \|\theta - \theta'\|\end{aligned}$$

where the second inequality uses Assumption 7 and the Lipschitz property of $\theta \mapsto \nabla_\theta w^\star_\theta(x)$ established in Corollary B.1.

Part (iii). For any $\phi, \phi' \in \Phi$,

$$|J(\phi, \theta) - J(\phi', \theta)| \le \mathbb{E}_x\big[\|f_\phi(x) - f_{\phi'}(x)\| \|w^\star_\theta(x)\|\big] \le B_{\mathcal{W}} L_{f_\phi} \|\phi - \phi'\|.$$

□

Proposition D.2 (Averaged bias gradient)**.** *Under Assumptions 5,6,7, and Corollary B.1,*

(i) *With $\eta_t = D_\Phi / (G_\mathcal{R} \sqrt{T})$,*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\big[\|\nabla_\theta B(\phi_t, \theta_t)\|^2\big] \;\le\; \frac{2 B^2_{\nabla w^\star_\theta} L^2_{f_\phi} D_\Phi G_\mathcal{R}}{\lambda \sqrt{T}}$$

(ii) *With $\eta_t = 1/(\lambda t)$,*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\big[\|\nabla_\theta B(\phi_t, \theta_t)\|^2\big] \;\le\; \frac{B^2_{\nabla w^\star_\theta} L^2_{f_\phi} G^2_\mathcal{R} (1 + \log T)}{\lambda^2 T}$$

Proof. Since $B(\phi, \theta) = J(\phi, \theta) - J(\phi^\star, \theta)$ by Assumption 5, then

$$\nabla_\theta B(\phi_t, \theta_t) = \mathbb{E}_x\big[(\nabla_\theta w^\star_{\theta_t}(x))^\top (f_{\phi_t}(x) - f^\star(x))\big].$$

By Jensen's inequality, Cauchy-Schwarz, Corollary B.1, and the Lipschitz parameterization (Assumption 7)

$$\|\nabla_\theta B(\phi_t, \theta_t)\|^2 \le B^2_{\nabla w^\star_\theta} \mathbb{E}_x\big[\|f_{\phi_t}(x) - f^\star(x)\|^2\big] \le B^2_{\nabla w^\star_\theta} L^2_{f_\phi} \|\phi_t - \phi^\star\|^2.$$

Under Assumption 5, $f^\star$ minimizes $\mathcal{R}_\theta(f)$ for every θ (Remark 2.1). Combined with λ-strong convexity (Assumption 6(iii)), this yields the quadratic-growth inequality

$$\tfrac{\lambda}{2} \|\phi_t - \phi^\star\|^2 \le \mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f^\star). \tag{18}$$

Combining the previous bound with (18) yields

$$\|\nabla_\theta B(\phi_t, \theta_t)\|^2 \le \frac{2 B^2_{\nabla w^\star_\theta} L^2_{f_\phi}}{\lambda} \big(\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f^\star)\big).$$

Averaging over $t = 1, \dots, T$ and applying Proposition 2.1 yields the two stated bounds. □

The hybrid gradient is conditionally biased, but only through the nuisance error, as the following identity formalizes.

Lemma D.3 (Hybrid gradient bias). *Under Assumptions 2,3,5, and the log-derivative identity for* g_t^{score},

$$\mathbb{E}\big[g_t^{\alpha} \mid \phi_t, \theta_t\big] = \nabla_\theta V^{\text{cost}}(\theta_t) + (1-\alpha)\,\nabla_\theta B(\phi_t, \theta_t) \tag{19}$$

Proof. By Assumption 3 and dominated convergence, we may exchange differentiation and expectation:

$$\nabla_\theta V^{\text{cost}}(\theta_t) = \mathbb{E}_{x,c,\hat{c}}[c^\top w^\star(\hat{c})\,\nabla_\theta \log p_{\theta_t}(\hat{c} \mid x)] = \mathbb{E}[g_t^{\text{score}} \mid \phi_t, \theta_t],$$

where the second equality uses $w_t = w^\star(\hat{c}_t)$ and $y_t = c_t^\top w_t$.

For the plug-in estimator, since $w_t = w^\star(\hat{c}_t)$ and applying the log-derivative identity,

$$\begin{aligned}
\mathbb{E}[g_t^{\text{plug-in}} \mid \phi_t, \theta_t] &= \mathbb{E}_{x,\hat{c}}\big[(w^\star(\hat{c})^\top f_{\phi_t}(x))\,\nabla_\theta \log p_{\theta_t}(\hat{c} \mid x)\big] \\
&= \mathbb{E}_x\big[f_{\phi_t}(x)^\top \mathbb{E}_{\hat{c}}\big[w^\star(\hat{c})\,\nabla_\theta \log p_{\theta_t}(\hat{c} \mid x)^\top\big]\big] \\
&= \mathbb{E}_x\big[f_{\phi_t}(x)^\top \nabla_\theta w^\star_{\theta_t}(x)\big] = \nabla_\theta J(\phi_t, \theta_t),
\end{aligned}$$

using Corollary B.1 for the third equality. Combining the two by linearity of expectation yields the claim. □

Assumption 8 (Gradient variances). *There exist* $\sigma_{\text{score}}, \sigma_{\text{plug-in}} < \infty$ *such that:*

$$\text{Var}\big[g_t^{\text{score}} \,\big|\, \phi_t, \theta_t\big] \le \sigma^2_{\text{score}}, \qquad \text{Var}\big[g_t^{\text{plug-in}} \,\big|\, \phi_t, \theta_t\big] \le \sigma^2_{\text{plug-in}}.$$

Lemma D.4 (One-Step Descent). *Under Assumptions 5–8, the iterates of the policy update satisfy*

$$\begin{aligned}
\mathbb{E}\big[V^{\text{cost}}(\theta_{t+1})\big] \le \mathbb{E}\big[V^{\text{cost}}(\theta_t)\big] &- \frac{\beta}{2}\mathbb{E}\big[\|\nabla_\theta V^{\text{cost}}(\theta_t)\|^2\big] \\
&+ \frac{\beta(1-\alpha)^2}{2}\mathbb{E}\big[\|\nabla_\theta B(\phi_t, \theta_t)\|^2\big] + \frac{L_\theta^J \beta^2}{2}((G^J)^2 + \sigma_\alpha^2).
\end{aligned} \tag{20}$$

with $\sigma_\alpha^2 := (\alpha\sigma_{\text{score}} + (1-\alpha)\sigma_{\text{plug-in}})^2$.

Proof of Lemma D.4. Since $V^{\text{cost}}(\theta) = J(\phi^\star, \theta)$ is L_θ^J-smooth in θ (Lemma D.1(ii)), the descent lemma applied to the update $\theta_{t+1} = \theta_t - \beta g_t^\alpha$ gives

$$V^{\text{cost}}(\theta_{t+1}) \le V^{\text{cost}}(\theta_t) + \big\langle \nabla_\theta V^{\text{cost}}(\theta_t),\, \theta_{t+1} - \theta_t \big\rangle + \frac{L_\theta^J}{2}\|\theta_{t+1} - \theta_t\|^2 \tag{21}$$

Substituting $\theta_{t+1} - \theta_t = -\beta g_t^\alpha$ and taking conditional expectation given (ϕ_t, θ_t),

$$\mathbb{E}\big[\langle \nabla_\theta V^{\text{cost}}(\theta_t),\, -\beta g_t^\alpha \rangle \mid \phi_t, \theta_t\big] = -\beta\big\langle \nabla_\theta V^{\text{cost}}(\theta_t),\, \mathbb{E}[g_t^\alpha \mid \phi_t, \theta_t]\big\rangle$$

Applying Lemma D.3, $\mathbb{E}[g_t^\alpha \mid \phi_t, \theta_t] = \nabla_\theta V^{\text{cost}}(\theta_t) + (1-\alpha)\nabla_\theta B(\phi_t, \theta_t)$, so

$$-\beta\big\langle \nabla_\theta V^{\text{cost}},\, \nabla_\theta V^{\text{cost}} + (1-\alpha)\nabla_\theta B\big\rangle = -\beta\|\nabla_\theta V^{\text{cost}}\|^2 - \beta\langle \nabla_\theta V^{\text{cost}},\, (1-\alpha)\nabla_\theta B\rangle$$

Applying the inequality $-\langle a, b\rangle \le \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$ to the cross term,

$$\mathbb{E}\big[\langle \nabla_\theta V^{\text{cost}},\, -\beta g_t^\alpha \rangle \mid \phi_t, \theta_t\big] \le -\frac{\beta}{2}\|\nabla_\theta V^{\text{cost}}(\theta_t)\|^2 + \frac{\beta(1-\alpha)^2}{2}\|\nabla_\theta B(\phi_t, \theta_t)\|^2 \tag{22}$$

We bound $\mathbb{E}[\|g_t^\alpha\|^2 \mid \phi_t, \theta_t]$ using the bias-variance decomposition

$$\mathbb{E}\big[\|g_t^\alpha\|^2 \mid \phi_t, \theta_t\big] = \text{Var}\big[g_t^\alpha \mid \phi_t, \theta_t\big] + \big\|\mathbb{E}\big[g_t^\alpha \mid \phi_t, \theta_t\big]\big\|^2 \tag{23}$$

For the variance term, write $g_t^\alpha = \alpha g_t^{\text{score}} + (1-\alpha) g_t^{\text{plug-in}}$. Expanding,

$$\begin{aligned}
\text{Var}\big[g_t^\alpha \mid \phi_t, \theta_t\big] = \alpha^2 \text{Var}\big[g_t^{\text{score}} \mid \phi_t, \theta_t\big] &+ (1-\alpha)^2 \text{Var}\big[g_t^{\text{plug-in}} \mid \phi_t, \theta_t\big] \\
&+ 2\alpha(1-\alpha)\text{Cov}\big[g_t^{\text{score}},\, g_t^{\text{plug-in}} \mid \phi_t, \theta_t\big]
\end{aligned}$$

We bound the covariance term via Cauchy-Schwarz

$$\left|\mathrm{Cov}\left[g_t^{\mathrm{score}},\, g_t^{\text{plug-in}} \mid \phi_t, \theta_t\right]\right| \le \sigma_{\mathrm{score}}\, \sigma_{\text{plug-in}}$$

where the second inequality uses Assumption 8. Substituting and completing the square,

$$\mathrm{Var}\left[g_t^{\alpha} \mid \phi_t, \theta_t\right] \le \left(\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\text{plug-in}}\right)^2 =: \sigma_\alpha^2 \tag{24}$$

For the squared mean term, by Lemma D.3, the triangle inequality, $\|\mathbb{E}[g_t^{\alpha} \mid \phi_t, \theta_t]\| \le \alpha\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\| + (1-\alpha)\|\nabla_\theta J(\phi_t, \theta_t)\| = \alpha\|\nabla_\theta J(\phi^\star, \theta_t)\| + (1-\alpha)\|\nabla_\theta J(\phi_t, \theta_t)\| \le G^J$ where the last inequality is given by Lemma D.1. Substituting into (23),

$$\mathbb{E}\left[\|g_t^{\alpha}\|^2 \mid \phi_t, \theta_t\right] \le \sigma_\alpha^2 + (G^J)^2 \tag{25}$$

Substituting (22) and (25) into (21) and taking full expectations,

$$\begin{aligned}\mathbb{E}\left[V^{\mathrm{cost}}(\theta_{t+1})\right] \le \mathbb{E}\left[V^{\mathrm{cost}}(\theta_t)\right] &- \frac{\beta}{2}\mathbb{E}\left[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\right] \\ &+ \frac{\beta(1-\alpha)^2}{2}\mathbb{E}\left[\|\nabla_\theta B(\phi_t, \theta_t)\|^2\right] + \frac{L_\theta^J \beta^2}{2}(\sigma_\alpha^2 + (G^J)^2)\end{aligned}$$

which is the claimed inequality. □

We are now in a position to state Theorem 2.2 with explicit constants and provide its proof. The result follows directly by combining the one-step descent inequality (Lemma D.4) with the averaged bias bound of Proposition D.2.

Theorem D.5 (Non-convex stationarity, plug-in variant)**.** *Let $V^\star := \inf_\theta V^{\mathrm{cost}}(\theta) > -\infty$ Under Assumptions 2–8, with step size $\beta = 1/(L_\theta^J\sqrt{T})$, Algorithm 1 satisfies,*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\right] \;\le\; \frac{2L_\theta^J\Delta_1 + (G^J)^2 + \sigma_\alpha^2}{\sqrt{T}} + (1-\alpha)^2 B_{\nabla w_\theta^\star}^2 L_{f_\phi}^2 \cdot \mathcal{B}_{\mathrm{bias}}(T), \tag{26}$$

where $\Delta_1 := V^{\mathrm{cost}}(\theta_1) - V^\star$, and the bias decay $\mathcal{B}_{\mathrm{bias}}(T)$ depends on the nuisance step size:

(i) *With $\eta_t = D_\Phi/(G_{\mathcal{R}}\sqrt{T})$:* $\mathcal{B}_{\mathrm{bias}}(T) = \frac{2D_\Phi G_{\mathcal{R}}}{\lambda\sqrt{T}}$.

(ii) *With $\eta_t = 1/(\lambda t)$:* $\mathcal{B}_{\mathrm{bias}}(T) = \frac{G_{\mathcal{R}}^2(1+\log T)}{\lambda^2 T}$.

Proof. Rearranging Lemma D.4,

$$\begin{aligned}\frac{\beta}{2}\mathbb{E}\left[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\right] \le \mathbb{E}\left[V^{\mathrm{cost}}(\theta_t)\right] &- \mathbb{E}\left[V^{\mathrm{cost}}(\theta_{t+1})\right] \\ &+ \frac{\beta(1-\alpha)^2}{2}\mathbb{E}\left[\|\nabla_\theta B(\phi_t, \theta_t)\|^2\right] + \frac{L_\theta^J\beta^2}{2}((G^J)^2 + \sigma_\alpha^2)\end{aligned}$$

Summing over $t = 1, \ldots, T$ and using $\mathbb{E}[V^{\mathrm{cost}}(\theta_{T+1})] \ge V^\star$,

$$\frac{\beta}{2}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\right] \le V^{\mathrm{cost}}(\theta_1) - V^\star + \frac{\beta(1-\alpha)^2}{2}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta B\|^2\right] + \frac{L_\theta^J\beta^2 T}{2}((G^J)^2 + \sigma_\alpha^2)$$

Dividing by $\beta T/2$ and setting $\beta = 1/(L_\theta^J\sqrt{T})$,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2\right] \le \frac{2L_\theta^J(V^{\mathrm{cost}}(\theta_1) - V^\star) + (G^J)^2 + \sigma_\alpha^2}{\sqrt{T}} + (1-\alpha)^2\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta B\|^2\right]$$

Applying Proposition D.2(i)-(ii) yields the results. □

E CONVERGENCE ANALYSIS: SURROGATE HYBRID VARIANT

This appendix extends the hybrid gradient framework to the surrogate setting, where the plug-in policy gradient is replaced by a differentiable proxy. A *surrogate cost function* $h^{\mathrm{sur}} : \mathbb{R}^{d_w} \times \mathbb{R}^{d_w} \to \mathbb{R}$ replaces the inner objective $c^\top w^\star(\hat{c})$ with a smooth, globally defined surrogate.

For a given policy θ, the plug-in surrogate objective is defined in Equation 6

$$J^{\mathrm{sur}}(\phi, \theta) := \mathbb{E}_x\big[\mathbb{E}_{\hat{c}\sim p_\theta(\cdot|x)}[h^{\mathrm{sur}}(f_\phi(x),\, \hat{c})]\big]$$

Realizability (Assumption 5) implies $V^{\mathrm{cost}}(\theta) = J(\phi^\star, \theta)$ as an exact identity, reducing the problem to the plug-in objective $J(\phi, \theta)$ and its surrogate $J^{\mathrm{sur}}(\phi, \theta)$. Our analysis therefore takes $J^{\mathrm{sur}}(\phi^\star, \theta)$ directly as the surrogate target.

The surrogate nuisance bias is entirely captured by the surrogate analog:

$$B^{\mathrm{sur}}(\phi, \theta) := J^{\mathrm{sur}}(\phi, \theta) - J^{\mathrm{sur}}(\phi^\star, \theta).$$

The bias vanishes when $f_\phi = f^\star$, isolating the contribution of nuisance error. Whenever the surrogate is faithful and ϕ is close to $\phi^\star$, a good policy under $J^{\mathrm{sur}}(\phi, \cdot)$ remains good under $J(\phi, \cdot)$, and therefore under V^{cost}.

Assumption 9 (Surrogate regularity)**.** *The surrogate $h^{\mathrm{sur}}(c, \hat{c})$ satisfies:*

(i) $\hat{c} \mapsto h^{\mathrm{sur}}(c, \hat{c})$ is continuously differentiable for every $c \in \mathbb{R}^{d_w}$, and there exists $G_{\mathrm{sur}} < \infty$ such that $\|\nabla_{\hat{c}} h^{\mathrm{sur}}(c, \hat{c})\| \le G_{\mathrm{sur}}$ almost surely.

(ii) There exist $L_c^{\mathrm{sur}}, L_{\hat{c}}^{\mathrm{sur}} < \infty$ such that for all $c, c', \hat{c}, \hat{c}' \in \mathbb{R}^{d_w}$,

$$\|\nabla_{\hat{c}} h^{\mathrm{sur}}(c, \hat{c}) - \nabla_{\hat{c}} h^{\mathrm{sur}}(c', \hat{c})\| \le L_c^{\mathrm{sur}} \|c - c'\|,$$
$$\|\nabla_{\hat{c}} h^{\mathrm{sur}}(c, \hat{c}) - \nabla_{\hat{c}} h^{\mathrm{sur}}(c, \hat{c}')\| \le L_{\hat{c}}^{\mathrm{sur}} \|\hat{c} - \hat{c}'\|.$$

To construct a tractable estimator of $\nabla_\theta J^{\mathrm{sur}}$ we assume that $p_\theta(\cdot \mid x)$ admits a reparameterization $\hat{c} = \mu_\theta(x, \varepsilon)$ with $\varepsilon \sim \nu$ independent of θ (Assumption 4). At iteration t, having sampled $\varepsilon_t \sim \nu$ and computed $\hat{c}_t = \mu_{\theta_t}(x_t, \varepsilon_t)$, the chain rule yields the surrogate gradient estimator

$$g_t^{\mathrm{sur}} := \big(\nabla_\theta \mu_{\theta_t}(x_t, \varepsilon_t)\big)^\top \nabla_{\hat{c}} h^{\mathrm{sur}}\big(f_{\phi_t}(x_t), \hat{c}_t\big)$$

Proposition E.1 (Unbiasedness of g_t^{sur})**.** *Under Assumptions 4 and 9,*

$$\mathbb{E}[g_t^{\mathrm{sur}} \mid \phi_t, \theta_t] = \nabla_\theta J^{\mathrm{sur}}(\phi_t, \theta_t)$$

Proof. By Assumption 4, $\hat{c}_t = \mu_{\theta_t}(x_t, \varepsilon_t)$ with $\varepsilon_t \sim \nu$ independent of θ_t, so the reparameterization trick and Assumptions 9(i) and 4 justify exchanging differentiation and expectation:

$$\nabla_\theta J^{\mathrm{sur}}(\phi_t, \theta_t) = \mathbb{E}_{x,\varepsilon}\Big[\big(\nabla_\theta \mu_{\theta_t}(x, \varepsilon)\big)^\top \nabla_{\hat{c}} h^{\mathrm{sur}}\big(f_{\phi_t}(x), \mu_{\theta_t}(x, \varepsilon)\big)\Big] = \mathbb{E}\big[g_t^{\mathrm{sur}} \mid \phi_t, \theta_t\big]. \qquad \square$$

The surrogate hybrid gradient estimator is $g_t^{\alpha,\mathrm{sur}} := \alpha\, g_t^{\mathrm{score}} + (1 - \alpha)\, g_t^{\mathrm{sur}}$.

We operate under the same regime as the plug-in policy case. Define the target objective

$$V^\alpha(\theta) := \alpha\, V^{\mathrm{cost}}(\theta) + (1 - \alpha)\, J^{\mathrm{sur}}(\phi^\star, \theta) \tag{27}$$

Assumption 10 (Surrogate gradient moments)**.** *There exist $\sigma_{\mathrm{sur}} < \infty$ such that* $\mathrm{Var}[g_t^{\mathrm{sur}} \mid \phi_t, \theta_t] \le \sigma_{\mathrm{sur}}^2$.

Assumption 11 (Surrogate gradient norm bound)**.** *There exists $G_{\mathrm{sur}}^J < \infty$ such that $\|\nabla_\theta J^{\mathrm{sur}}(\phi, \theta)\| \le G_{\mathrm{sur}}^J$ for all $\phi \in \Phi$, $\theta \in \mathbb{R}^{d_\theta}$. Under Assumptions 4–9, this holds with $G_{\mathrm{sur}}^J := G_\mu G_{\mathrm{sur}}$.*

Let $\tilde{G}_\alpha := \alpha G^J + (1 - \alpha) G_{\mathrm{sur}}^J$. By Lemma D.1(i) and Assumption 11, $\tilde{G}_\alpha$ bounds the norm of $\mathbb{E}[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t]$.

Lemma E.2 (Surrogate hybrid gradient bias)**.** *Under Assumptions 2,3,4,5,9 and the log-derivative identity for g_t^{score},*

$$\mathbb{E}\big[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t\big] = \nabla_\theta V^\alpha(\theta_t) + (1 - \alpha)\, \nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)$$

Proof. By the log-derivative identity, $\mathbb{E}[g_t^{\mathrm{score}} \mid \phi_t, \theta_t] = \nabla_\theta V^{\mathrm{cost}}(\theta_t)$. Unbiasedness of the reparameterization estimator gives $\mathbb{E}[g_t^{\mathrm{sur}} \mid \phi_t, \theta_t] = \nabla_\theta J^{\mathrm{sur}}(\phi_t, \theta_t) = \nabla_\theta J^{\mathrm{sur}}(\phi^\star, \theta_t) + \nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)$. Linearity of expectation yields the claim. □

Lemma E.3 (Smoothness of V^α)**.** *Under Assumptions 5,9 and 4, $J^{\mathrm{sur}}(\phi^\star, \cdot)$ is L_θ^{sur}-smooth with $L_\theta^{\mathrm{sur}} := L_\mu G_{\mathrm{sur}} + G_\mu^2 L_{\hat{c}}^{\mathrm{sur}}$. Consequently V^α is L_θ^α-smooth with $L_\theta^\alpha := \alpha L_\theta^J + (1-\alpha) L_\theta^{\mathrm{sur}}$.*

Proof. We show $J^{\mathrm{sur}}(\phi^\star, \cdot)$ is L_θ^{sur}-smooth, the smoothness of V^α then follows because a convex combination of smooth functions is smooth. By Assumption 4, we may write

$$\nabla_\theta J^{\mathrm{sur}}(\phi^\star, \theta) = \mathbb{E}_{x,\varepsilon}\left[(\nabla_\theta \mu_\theta(x,\varepsilon))^\top \nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \mu_\theta(x,\varepsilon))\right].$$

For any $\theta, \theta' \in \mathbb{R}^{d_\theta}$, denote $\hat{c} = \mu_\theta(x,\varepsilon)$ and $\hat{c}' = \mu_{\theta'}(x,\varepsilon)$. Adding and subtracting $(\nabla_\theta \mu_{\theta'}(x,\varepsilon))^\top \nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \hat{c}')$ and applying the triangle inequality,

$$\begin{aligned}
&\|\nabla_\theta J^{\mathrm{sur}}(\phi^\star, \theta) - \nabla_\theta J^{\mathrm{sur}}(\phi^\star, \theta')\| \\
&\quad \le \mathbb{E}_{x,\varepsilon}[\|\nabla_\theta \mu_\theta(x,\varepsilon) - \nabla_\theta \mu_{\theta'}(x,\varepsilon)\| \, \|\nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \hat{c})\|] \\
&\qquad + \mathbb{E}_{x,\varepsilon}[\|\nabla_\theta \mu_{\theta'}(x,\varepsilon)\| \, \|\nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \hat{c}) - \nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \hat{c}')\|] \\
&\quad \le L_\mu G_{\mathrm{sur}} \, \|\theta - \theta'\| + G_\mu L_{\hat{c}}^{\mathrm{sur}} \, \mathbb{E}_{x,\varepsilon}[\|\hat{c} - \hat{c}'\|] \\
&\quad \le \left(L_\mu G_{\mathrm{sur}} + G_\mu^2 L_{\hat{c}}^{\mathrm{sur}}\right) \|\theta - \theta'\|,
\end{aligned}$$

where the second inequality uses Assumption 4 ($\|\nabla_\theta \mu\| \le G_\mu$, L_μ-Lipschitz) and Assumption 9(i)-(ii); the third inequality uses $\|\hat{c} - \hat{c}'\| = \|\mu_\theta(x,\varepsilon) - \mu_{\theta'}(x,\varepsilon)\| \le G_\mu \|\theta - \theta'\|$ from Assumption 4. This proves smoothness of $J^{\mathrm{sur}}(\phi^\star, \cdot)$. Since $V^{\mathrm{cost}} = J(\phi^\star, \cdot)$ is L_θ^J-smooth (Lemma D.1(ii)) and V^α is a convex combination, V^α is L_θ^α-smooth with $L_\theta^\alpha := \alpha L_\theta^J + (1-\alpha) L_\theta^{\mathrm{sur}}$. □

Proposition E.4 (Averaged surrogate bias gradient)**.** *Under Assumptions 5–6,9–10, with step size $\eta_t = 1/(\lambda t)$ in the nuisance update,*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\|\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)\|^2\right] \le \frac{(L_c^{\mathrm{sur}})^2 G_\mu^2 \, L_{f_\phi}^2 \, G_{\mathcal{R}}^2 (1 + \log T)}{\lambda^2 T} = \mathcal{O}\left(\frac{\log T}{T}\right)$$

Proof. Differentiating under the expectation (justified by Assumptions 9(i) and 4),

$$\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t) = \mathbb{E}_{x,\varepsilon}\Big[(\nabla_\theta \mu_{\theta_t}(x,\varepsilon))^\top \left(\nabla_{\hat{c}} h^{\mathrm{sur}}(f_{\phi_t}(x), \hat{c}) - \nabla_{\hat{c}} h^{\mathrm{sur}}(f^\star(x), \hat{c})\right)\Big]$$

Jensen's inequality, Cauchy-Schwarz, $\|\nabla_\theta \mu\| \le G_\mu$, and Assumption 9(ii) give

$$\|\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)\|^2 \le (L_c^{\mathrm{sur}})^2 \, G_\mu^2 \, \mathbb{E}_x\left[\|f_{\phi_t}(x) - f^\star(x)\|^2\right] \tag{28}$$

By strong convexity $\mathbb{E}_x[\|f_{\phi_t}(x) - f^\star(x)\|^2] \le \frac{2L_{f_\phi}^2}{\lambda} \, \mathbb{E}[\mathcal{R}_{\theta_t}(f_{\phi_t}) - \mathcal{R}_{\theta_t}(f^\star)]$. Averaging (28) over t and applying Proposition 2.1(ii) yields the claim. □

Lemma E.5 (One-Step Descent, Surrogate Variant)**.** *Under Assumptions 5–6,9–10, the iterates of the hybrid update $\theta_{t+1} = \theta_t - \beta g_t^{\alpha,\mathrm{sur}}$ satisfy*

$$\begin{aligned}
\mathbb{E}[V^\alpha(\theta_{t+1})] - \mathbb{E}[V^\alpha(\theta_t)] \le &-\frac{\beta}{2}\mathbb{E}\left[\|\nabla V^\alpha(\theta_t)\|^2\right] \\
&+ \frac{\beta(1-\alpha)^2}{2}\mathbb{E}\left[\|\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)\|^2\right] \\
&+ \frac{L_\theta^\alpha \beta^2}{2}(\tilde{G}_\alpha^2 + \tilde{\sigma}_\alpha^2).
\end{aligned} \tag{29}$$

with $\tilde{\sigma}_\alpha^2 := (\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\mathrm{sur}})^2$ and $\tilde{G}_\alpha := \alpha G^J + (1-\alpha) G_{\mathrm{sur}}^J$.

Proof. By Lemma E.3, V^α is L_θ^α-smooth, so the descent lemma yields

$$V^\alpha(\theta_{t+1}) \le V^\alpha(\theta_t) + \langle \nabla V^\alpha(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L_\theta^\alpha}{2}\|\theta_{t+1} - \theta_t\|^2$$

Substituting $\theta_{t+1} - \theta_t = -\beta g_t^{\alpha,\mathrm{sur}}$ and taking conditional expectation given (ϕ_t, θ_t), Lemma E.2 gives

$$\mathbb{E}[\langle \nabla V^\alpha, -\beta g_t^{\alpha,\mathrm{sur}} \rangle \mid \phi_t, \theta_t] = -\beta \|\nabla V^\alpha(\theta_t)\|^2 - \beta \langle \nabla V^\alpha(\theta_t), (1-\alpha)\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t) \rangle$$

Applying Young's inequality to the cross term,

$$\mathbb{E}[\langle \nabla V^\alpha, -\beta g_t^{\alpha,\mathrm{sur}} \rangle \mid \phi_t, \theta_t] \le -\frac{\beta}{2}\|\nabla V^\alpha(\theta_t)\|^2 + \frac{\beta(1-\alpha)^2}{2}\|\nabla_\theta B^{\mathrm{sur}}(\phi_t, \theta_t)\|^2 \tag{30}$$

For the quadratic term, the bias-variance decomposition gives $\mathbb{E}[\|g_t^{\alpha,\mathrm{sur}}\|^2 \mid \phi_t, \theta_t] = \mathrm{Var}[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t] + \|\mathbb{E}[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t]\|^2$. By the same Cauchy-Schwarz argument as in the proof of Lemma D.4 (applied with σ_{sur} in place of $\sigma_{\text{plug-in}}$), and linearity of expectation

$$\mathrm{Var}[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t] \le (\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\mathrm{sur}})^2 = \tilde{\sigma}_\alpha^2, \quad \|\mathbb{E}[g_t^{\alpha,\mathrm{sur}} \mid \phi_t, \theta_t]\|^2 \le \tilde{G}_\alpha^2$$

with $\tilde{G}_\alpha := \alpha G^J + (1-\alpha)G_{\mathrm{sur}}^J$. Substituting into the descent lemma and taking full expectations gives (29). □

We restate Theorem 2.3 with explicit constants. The result follows by the same argument as Theorem 2.2, replacing Lemma D.4 with Lemma E.5 and Proposition D.2 with Proposition E.4.

Theorem E.6 (Non-convex stationarity, surrogate variant)**.** *Let $V^{\alpha,\star} := \inf_\theta V^\alpha(\theta)$ where $V^\alpha(\theta) := \alpha\, V^{\mathrm{cost}}(\theta) + (1-\alpha)\, J^{\mathrm{sur}}(\phi^\star, \theta)$. Running Algorithm 1 with $\beta = 1/(L_\theta^\alpha\sqrt{T})$ and $\eta_t = 1/(\lambda t)$ yields*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla V^\alpha(\theta_t)\|^2\big] \;\le\; \frac{2L_\theta^\alpha\Delta_1^\alpha + \tilde{G}_\alpha^2 + \tilde{\sigma}_\alpha^2}{\sqrt{T}} + \frac{(1-\alpha)^2(L_c^{\mathrm{sur}})^2 G_\mu^2\, L_{f_\phi}^2 G_{\mathcal{R}}^2(1+\log T)}{\lambda^2 T} \tag{31}$$

with $\Delta_1^\alpha = V^\alpha(\theta_1) - V^{\alpha,\star}$, $\tilde{\sigma}_\alpha^2 := (\alpha\sigma_{\mathrm{score}} + (1-\alpha)\sigma_{\mathrm{sur}})^2$ and $\tilde{G}_\alpha := \alpha G^J + (1-\alpha)G_{\mathrm{sur}}^J$.

Proof of Theorem 2.3. Following the same proof as Theorem 2.2, rearranging Lemma E.5, summing over $t = 1, \dots, T$, telescoping, and using $\mathbb{E}[V^\alpha(\theta_{T+1})] \ge V^{\alpha,\star}$,

$$\frac{\beta}{2}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla V^\alpha(\theta_t)\|^2\big] \le V^\alpha(\theta_1) - V^{\alpha,\star} + \frac{\beta(1-\alpha)^2}{2}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla_\theta B^{\mathrm{sur}}\|^2\big] + \frac{L_\theta^\alpha\beta^2 T}{2}(\tilde{G}_\alpha^2 + \tilde{\sigma}_\alpha^2)$$

Dividing by $\beta T/2$, substituting $\beta = 1/(L_\theta^\alpha\sqrt{T})$, and applying Proposition E.4 yields (31). □

F Benchmark formulations and data

F.1 Top-k selection

This benchmark models the selection of exactly k items from the candidate set $\mathcal{I}_{\mathrm{top}} = \{1, \dots, d\}$. The binary decision variable w_i indicates whether item i is selected. Given predicted item costs $\hat{c}_t$, the top-k oracle solves

$$\min_{w \in \{0,1\}^d} \hat{c}_t^\top w \quad \text{s.t.} \quad \mathbf{1}^\top w = k.$$

The default top-k configuration has $d = 15$ items, cardinality $k = 2$, and context dimension $d_x = 5$. Contexts are sampled i.i.d. according to $x_t \sim \mathcal{N}(0, I_{d_x})$. At environment construction time, the generator draws and fixes a binary coefficient matrix $\omega_{\mathrm{top}} \in \{0,1\}^{d \times d_x}$ with i.i.d. Bernoulli(0.5) entries. Let deg denote the polynomial degree parameter and $\bar{\xi}$ the multiplicative-noise radius. At iteration t, it forms a latent vector $u_t \in \mathbb{R}^d$ as $u_t = [1 + (1 + \omega_{\mathrm{top}} x_t/\sqrt{d_x})^{\mathrm{deg}}] \odot \xi_t$, where $\xi_{t,j} \sim \mathrm{Unif}[1-\bar{\xi},\, 1+\bar{\xi}]$ when multiplicative noise is enabled, while $\xi_{t,j} = 1$ otherwise. The realized cost vector is then $c_t = u_t$. Because x_t is Gaussian, these costs, and the shortest-path costs below, have unbounded support, whereas the analysis assumes bounded costs (Assumption 2).

F.2 SHORTEST PATH

This benchmark uses a fixed directed acyclic grid graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with right/down edges, where $|\mathcal{V}| = 25$ and $|\mathcal{E}| = q = 40$. The binary decision variable w_e indicates whether edge e is on the selected path. Let A denote the node–arc incidence matrix and $b_{sd} \in \mathbb{R}^{|\mathcal{V}|}$ the source–destination balance vector ($+1$ at the source, -1 at the destination, 0 elsewhere). Given predicted edge costs $\hat{c}_t$, the shortest-path oracle solves

$$\min_{w \in \{0,1\}^q} \ \hat{c}_t^\top w \quad \text{s.t.} \quad Aw = b_{sd}.$$

The default shortest-path configuration has context dimension $d_x = 10$ and $q = 40$ edges over the fixed 5×5 right/down DAG described in Section F.2. Contexts are sampled i.i.d. as $x_t \sim \mathcal{N}(0, I_{d_x})$. The generator draws an independent binary coefficient matrix $\omega_{\text{sp}} \in \{0,1\}^{q \times d_x}$ and forms $u_t \in \mathbb{R}^q$ by the same polynomial construction as in the top-k instance, with ω_{sp} replacing ω_{top}. The generated vector is used directly as the realized edge-cost vector, $c_t = u_t$.

F.3 COMBINATORIAL PRICING

This benchmark assigns one of K_{price} discrete price levels to each of n products. Let $w_{i\ell}$ indicate whether product i is assigned price level ℓ, let $L_{\text{promo}} \le K_{\text{price}}$ be the number of lowest price tiers counted as promotional, and let B_{promo} be the promotion budget. Given predicted product–price costs $\hat{c}_t$, the induced decision solves

$$\min_{w \in \{0,1\}^{nK_{\text{price}}}} \ \hat{c}_t^\top w \quad \text{s.t.} \quad \sum_{\ell=1}^{K_{\text{price}}} w_{i\ell} = 1 \ \forall i, \quad \sum_{i=1}^{n} \sum_{\ell=1}^{L_{\text{promo}}} w_{i\ell} \le B_{\text{promo}}.$$

Conditional on x_t, latent demand is sampled for every product–price pair; the action determines which entries are monetized, and the benchmark is written as cost minimization with cost equal to negative realized revenue.

The default pricing configuration has $n = 20$ products, $K_{\text{price}} = 4$ normalized price levels $\tilde{\pi}_\ell$ equally spaced in $[0.1, 0.9]$, promotion threshold $L_{\text{promo}} = 2$, budget $B_{\text{promo}} = 7$, and contexts $x_t \sim \mathcal{N}(0, I_{d_x})$ with $d_x = 25$. The generator fixes unit directions b_i for each product and b_0 for the market. Demand follows a logit purchase model with willingness to pay $v_i(x) = 0.65 + 0.5 \tanh\big(0.75\,[\,0.4\, g_{\deg}(b_0^\top x) + 0.6\, g_{\deg}(b_i^\top x)\,]\big)$, where $g_{\deg}(z) = \text{sign}(z)(|z|/0.674)^{\deg}$ keeps its typical size fixed as the degree changes. The mean demand for product i at level ℓ is $\mu_{i\ell}(x) = 3\lambda_\ell\, \sigma\big((v_i(x) - \tilde{\pi}_\ell)/0.1\big)$, with σ the logistic function and a promotion lift $\lambda_\ell = 1.25$ on the promotional levels ($\lambda_\ell = 1$ otherwise). Demand $D_{i\ell,t}$ is Poisson with this mean times a common lognormal shock of scale 0.1, the cost vector has entries $c_{i\ell,t} = -\tilde{\pi}_\ell D_{i\ell,t}$, and the bandit feedback is the realized cost $v_t = c_t^\top w_t$. The paper uses $\deg \in \{2, 3, 4, 6\}$, with default $\deg = 6$.

F.4 ENERGY SCHEDULING

This benchmark schedules tasks on machines under day-ahead electricity prices, following Mandi et al. (2024) with empirical SEMO data. Each iteration t corresponds to one day. Each task j has duration d_j slots, and each resource u has machine-m capacity $\bar{\rho}_{mu}$ and per-task usage ρ_{ju}. The binary decision $w_{jms} \in \{0,1\}$ indicates that task j starts on machine m at slot s, with $w \in \{0,1\}^{d_w}$ indexing feasible (task, machine, start-slot) triples. Aggregating the day's per-slot prices over each task's run window yields an iteration-t cost vector $c_t \in \mathbb{R}^{d_w}$ that depends linearly on the price profile. Given a predicted cost $\hat{c}_t$, the scheduling oracle solves

$$\min_{w \in \{0,1\}^{d_w}} \ \hat{c}_t^\top w \ \text{ s.t. } \sum_{m,s} w_{jms} = 1 \ \forall j, \ \sum_{j} \rho_{ju} \sum_{s : s \le s' < s + d_j} w_{jms} \le \bar{\rho}_{mu} \ \forall (m, u, s').$$

The energy-scheduling benchmark of Section F.4 follows the decision-focused learning benchmark suite of Mandi et al. (2024) and uses empirical day-level SEMO price data. The context is a 48×8 slot-feature matrix, and $\mathcal{J}$, $\mathcal{M}$, $\mathcal{U}$, $\mathcal{T}_{\text{slot}} = \{1, \ldots, 48\}$ index tasks, machines, resources, and slots. With per-task power P_j and slot length δ_{slot}, the iteration-t cost vector $c_t \in \mathbb{R}^{d_w}$ of

Section F.4 has components $c_{jms,t} := P_j \, \delta_{\text{slot}} \sum_{s'=s}^{s+d_j-1} \pi^{\text{el}}_{t,s'}$, giving the energy cost of task j when started on machine m at slot s under price profile π^{el}_t, where $\pi^{\text{el}}_{t,s'}$ is the electricity price in slot s' of day t. The scheduling instance is fixed within a run, with feasible start-slot sets $\mathcal{T}_j = \{s : a_j \le s,\ s + d_j - 1 \le b_j\}$ given by per-task earliest start a_j and latest end b_j; remaining instance specifics (covariate definitions, exact task counts, and resource limits) follow Mandi et al. (2024). This benchmark uses chronological train/test splits over days rather than freshly sampled Gaussian contexts. Because prices are observed data, the mean cost $f^\star(x_t)$ in the regret is the realized cost vector c_t on this benchmark.

F.5 Oracles and feedback modes

Optimization oracles. All four oracles are solved exactly: top-k by sorting predicted item costs, shortest path and combinatorial pricing by dynamic programming, and energy scheduling as a mixed-integer program with Gurobi.

Feedback modes. All four benchmarks support pure bandit, semi-bandit, and full-information feedback in the cost-vector representation $c_t \in \mathbb{R}^{d_w}$ that matches Assumption 1. The three feedback modes share the same (H, e, d_v) across benchmarks: pure bandit reveals the scalar realized cost $v_t = c_t^\top w_t$ via $H(w) = w^\top$, $e(w) = 1$, $d_v = 1$; semi-bandit reveals $v_t = \text{diag}(w_t)\, c_t$, the cost contribution of each action-selected coordinate, via $H(w) = \text{diag}(w)$, $e(w) = \mathbf{1}$, $d_v = d_w$; full information reveals $v_t = c_t$ via $H(w) = I$, $e(w) = w$, $d_v = d_w$. What differs across benchmarks is only what each coordinate of c_t represents: per-item costs (top-k), per-edge costs (shortest path), per-(product, price-level) realized demand contributions (pricing), and per-(task, machine, start-slot) energy costs (energy). On the energy benchmark, semi-bandit feedback exposes these per-(task, machine, start-slot) entries of the lifted c_t, not the original 48-slot price profile. Under full information, DFHPG plugs the observed cost vector c_t directly into the plug-in gradient in place of the nuisance estimate $f_{\phi_t}(x_t)$, and the nuisance model is not used.

G Implementation details of DFHPG

This appendix specifies the components of the reported DFHPG configurations: the variance-reduction baselines, the mixing-weight schedule, the surrogate losses, and the generative cost models.

G.1 Score function variance-reduction baselines

The score function component in Algorithm 1 uses the realized scalar cost $y_t = e(w_t)^\top v_t = c_t^\top w_t$ as the weight on $\nabla_\theta \log p_{\theta_t}(\hat{c}_t \mid x_t)$. In experiments we replace y_t with $y_t - b_t$ for some baseline b_t,

$$g_t^{\text{score}} = (y_t - b_t)\, \nabla_\theta \log p_{\theta_t}(\hat{c}_t \mid x_t),$$

which preserves unbiasedness whenever b_t is independent of $\hat{c}_t$ given x_t and typically reduces variance. We consider two baselines.

Moving-average baseline. A scalar running average of past realized costs y_t. Initialize $\bar{y}_1 = 0$ and update

$$b_t^{\text{ma}} = \bar{y}_t, \qquad \bar{y}_{t+1} = \psi\, \bar{y}_t + (1 - \psi)\, y_t,$$

with default momentum $\psi = 0.95$. This baseline adds no oracle call but ignores the context x_t.

Nuisance-induced baseline. A context-dependent baseline computed from the nuisance estimate. Let

$$w_t^\phi := w^\star(f_{\phi_t}(x_t)) \in \arg\min_{w \in \mathcal{W}} f_{\phi_t}(x_t)^\top w$$

denote the nuisance-induced optimal decision, and define

$$b_t^{\text{ni}} = f_{\phi_t}(x_t)^\top w_t^\phi,$$

the nuisance's own predicted optimum at x_t. When the implemented decision w_t matches w_t^ϕ, the gradient weight $y_t - b_t^{\text{ni}}$ collapses to the residual $y_t - f_{\phi_t}(x_t)^\top w_t$. When w_t is worse than w_t^ϕ

under the nuisance, the nuisance-predicted part $f_{\phi_t}(x_t)^\top w_t - b_t^{\mathrm{ni}}$ of the weight is nonnegative, so on average the gradient descent step reduces the probability of generating $\hat{c}_t$. This choice uses the context but adds one oracle solve per iteration. Since b_t^{ni} depends only on x_t and past data, a miscalibrated nuisance affects the variance of the score function gradient but not its unbiasedness.

G.2 Adaptive mixing-weight schedule

Algorithm 1 treats the mixing parameter α as a fixed scalar; in our experiments we instead use a time-varying α_t, applied in the hybrid gradient $g_t^{\alpha_t} = \alpha_t\, g_t^{\mathrm{score}} + (1-\alpha_t)\, g_t^{\text{plug-in}}$ (and analogously with g_t^{sur} in the surrogate variant). The schedule decays α_t from $\alpha_{\max}$ toward $\alpha_{\min}$, weighting the unbiased score function gradient more heavily early on (when the nuisance estimate is unreliable) and the lower-variance plug-in gradient later (as the nuisance estimate stabilizes). We use two variants that differ only in how the decay is gated. Both gates are specified in absolute iterations rather than as a fraction of T to avoid leaking the horizon into the schedule.

Fixed-warmup variant. For the first T_{warm} iterations (default $T_{\mathrm{warm}} = 100$), the target is held at $\alpha_t^\star = \alpha_{\max}$; afterwards $\alpha_t^\star = \alpha_{\min}$.

Exponential-gate variant. Used by the energy configurations and the pricing CNF configuration. The target blends $\alpha_{\max}$ and $\alpha_{\min}$ via an exponentially decaying weight,

$$\gamma_t = \exp(-(t-1)/\tau_{\mathrm{sched}})\,, \qquad \alpha_t^\star = \gamma_t\, \alpha_{\max} + (1-\gamma_t)\, \alpha_{\min},$$

where τ_{sched} controls the decay timescale (smaller τ_{sched} moves to the $\alpha_{\min}$ regime faster).

Smoothed update. In both variants, the actual mixing weight applies smoothing with rate η_α (default 0.05) and re-clips,

$$\alpha_t = \mathrm{clip}((1-\eta_\alpha)\, \alpha_{t-1} + \eta_\alpha\, \alpha_t^\star,\ \alpha_{\min},\ \alpha_{\max})\,.$$

Remark. Both variants use $\alpha_{\min}$ as a fixed post-warmup target. More adaptive heuristics are straightforward to substitute: for example, $\alpha_{\min}$ can be replaced by a reliability-driven target that responds to the running prediction error of the nuisance estimate (so that α_t rises again whenever the nuisance loses accuracy).

G.3 Surrogate-loss definitions

We follow the gradient form of each surrogate from the reference implementations of PyEPO (Tang & Khalil, 2024); pool-based variants follow the cumulative-pool formulation of the PredOpt-Benchmarks suite of Mandi et al. (2024). Throughout, $\hat{c}$ is the predicted cost vector, c the target cost (the true cost in offline DFL; the nuisance estimate $f_\phi(x)$ in our hybrid plug-in update), $w^\star(\cdot) \in \arg\min_{w\in\mathcal{W}}(\cdot)^\top w$ the linear-optimization oracle, $w_{\mathrm{ptb}}^\star(c) := \mathbb{E}_\xi[w^\star(c+\sigma\xi)]$ a perturbed-oracle expectation with $\xi \sim \mathcal{N}(0, I)$ and perturbation scale $\sigma > 0$, and $\mathcal{L}_{\mathrm{task}}(w; c) = c^\top w$ the regret-style task loss. In the tables, $\beta_{\mathrm{mse}} \in [0,1]$ is the weight on the mean-squared-error (MSE) term, $\lambda_{\mathrm{ptb}} > 0$ is the interpolation step of DBB and IMLE, and M is the number of perturbation samples in DPO. DPO, NCE, MAP, LTR, and EMA stand for differentiable perturbed optimizer, noise-contrastive estimation, maximum a posteriori, learning to rank, and exponential moving average. All subgradients below are stated with respect to $\hat{c}$ in the cost-minimization convention; the parameter update applies $\hat{c} \leftarrow \hat{c} - \eta\, \nabla_{\hat{c}} h^{\mathrm{sur}}$.

Oracle-call surrogates. These call $w^\star$ once or twice per gradient evaluation.

Loss	Subgradient $\nabla_{\hat{c}}\, h^{\mathrm{sur}}(c, \hat{c})$
SPO+ (Elmachtoub & Grigas, 2022)	$2\big(w^\star(c) - w^\star(2\hat{c} - c)\big)$
MSE+SPO+	$\beta_{\mathrm{mse}}(\hat{c} - c) + (1 - \beta_{\mathrm{mse}})\, \nabla_{\hat{c}} h_{\mathrm{SPO+}}$
DBB (Vlastelica Pogančić et al., 2020)	$\big(w^\star(\hat{c} + \lambda_{\mathrm{ptb}} c) - w^\star(\hat{c})\big)/\lambda_{\mathrm{ptb}}$
IMLE (Niepert et al., 2021)	$\big(w^\star_{\mathrm{ptb}}(\hat{c} + \lambda_{\mathrm{ptb}} c) - w^\star_{\mathrm{ptb}}(\hat{c})\big)/\lambda_{\mathrm{ptb}}$
AIMLE (Minervini et al., 2023)	IMLE with adaptive $\lambda_{\mathrm{ptb}} = \kappa_{\mathrm{aim}}\, \|\hat{c}\|/\|c\|$, κ_{aim} updated by EMA on grad sparsity
NID (Sahoo et al., 2023)	$-c$ (negative-identity straight-through)
PFYL (Berthet et al., 2020)	$w^\star(c) - w^\star_{\mathrm{ptb}}(\hat{c})$
DPO (Berthet et al., 2020)	$\frac{1}{M\sigma} \sum_{i=1}^{M} \xi_i \langle w^\star(\hat{c} + \sigma\xi_i),\, c\rangle,\ \xi_i \sim \mathcal{N}(0, I)$
PG (Gupta & Huang, 2024)	$\big(w^\star(\hat{c}) - w^\star(\hat{c} - \sigma c)\big)/\sigma$

Pool-based surrogates. Let $\mathcal{P}_t$ denote the cumulative pool of feasible solutions at iteration t, grown online by appending $w^\star(\hat{c}_t)$ and the solution $w^\star(c)$ for the target cost c at each step. Let $w_b := \arg\min_{w \in \mathcal{P}_t} c^\top w$ denote the pool's best solution under the target cost and $\mathcal{P}_t^- := \mathcal{P}_t \setminus \{w_b\}$ the rest of the pool; let $\delta > 0$ be the LTR margin and $\tau_{\mathrm{lw}} > 0$ the listwise temperature.

Loss	Subgradient $\nabla_{\hat{c}}\, h^{\mathrm{sur}}(c, \hat{c})$
NCE (Mulamba et al., 2021)	$w^\star(c) - \frac{1}{\lvert\mathcal{P}_t\rvert} \sum_{w \in \mathcal{P}_t} w$
contrastiveMAP (Mulamba et al., 2021)	$w^\star(c) - \arg\min_{w \in \mathcal{P}_t} \hat{c}^\top w$
pointwiseLTR (Mandi et al., 2022)	$\frac{2}{\lvert\mathcal{P}_t\rvert} \sum_{w \in \mathcal{P}_t} \big((\hat{c} - c)^\top w\big) w$
pairwiseDiff (Mandi et al., 2022)	$\frac{2}{\lvert\mathcal{P}_t^-\rvert} \sum_{r \in \mathcal{P}_t^-} \Delta_{br}(w_b - w_r),\ \Delta_{br} = (\hat{c} - c)^\top (w_b - w_r)$
pairwiseLTR (Mandi et al., 2022)	$\frac{1}{\lvert\mathcal{P}_t^-\rvert} \sum_{r \in \mathcal{P}_t^-\,:\, \hat{c}^\top w_b > \hat{c}^\top w_r - \delta} (w_b - w_r)$ (max-margin hinge)
listwiseLTR (Mandi et al., 2022)	$\frac{1}{\tau_{\mathrm{lw}}} \big(\sigma_{\mathrm{soft}}(-c^\top \mathcal{P}_t/\tau_{\mathrm{lw}}) - \sigma_{\mathrm{soft}}(-\hat{c}^\top \mathcal{P}_t/\tau_{\mathrm{lw}})\big)\, \mathcal{P}_t$
NCE_c (Mulamba et al., 2021)	Same subgradient as NCE; cached/centered pool variant
MAP_c (MAP-C) (Mulamba et al., 2021)	$w^\star(c) - \arg\min_{w \in \mathcal{P}_t} (\hat{c} - c)^\top w$ (cached MAP variant)
SPOCaching (Mulamba et al., 2021)	$w^\star(c) - \arg\min_{w \in \mathcal{P}_t} (2\hat{c} - c)^\top w$

Here σ_{soft} denotes softmax and $\hat{c}^\top \mathcal{P}_t \in \mathbb{R}^{|\mathcal{P}_t|}$ stacks the predicted-cost utilities of pool entries. The negative signs on the softmax arguments convert cost minimization to score maximization for the listwise ranking step. The heatmaps use implementation labels: SPOPlus and MSE+SPOPlus correspond to the SPO+ and MSE+SPO+ rows above, and MAP-C is the selected-configuration spelling of MAP_c.

Use in the hybrid update. With the reparameterization $\hat{c} = \mu_\theta(x, \varepsilon)$ of Assumption 4, any surrogate above enters the policy parameters via the chain rule $g_t^{\mathrm{sur}} = (\nabla_\theta \mu_{\theta_t}(x_t, \varepsilon_t))^\top \nabla_{\hat{c}} h^{\mathrm{sur}}(f_{\phi_t}(x_t), \hat{c}_t)$, i.e., the nuisance estimate $f_\phi(x)$ supplies the target cost c in each formula above. Theorem 2.3 covers the convergence of the resulting hybrid update for any surrogate whose subgradient satisfies the regularity bounds of Assumption 9.

G.4 Generative parameterizations

The CNF variant uses a two-layer conditional RealNVP flow (Dinh et al., 2017) with 128-dimensional coupling networks; the DDPM variant uses a conditional denoiser (Ho et al., 2020) with a 128-dimensional hidden layer.

Conditional normalizing-flow score function. For a conditional normalizing flow with tractable density $p_\theta(\cdot \mid x)$, the score function gradient uses the exact log-density,

$$g_{t,\mathrm{CNF}}^{\mathrm{score}} = y_t\, \nabla_\theta \log p_{\theta_t}(\hat{c}_t \mid x_t)\,,$$

Conditional diffusion score function. For a conditional diffusion model $p_\theta(\cdot \mid x)$, the log-density is intractable, so the score function gradient is approximated using an evidence lower bound (ELBO) proxy $\mathrm{ELBO}_\theta(c, x)$ for $\log p_\theta(c \mid x)$ (Ho et al., 2020),

$$g_{t,\mathrm{diff}}^{\mathrm{score}} = y_t\, \nabla_\theta \mathrm{ELBO}_{\theta_t}(\hat{c}_t, x_t)\,,$$

Because ELBO_θ is a biased proxy for $\log p_\theta$, the diffusion variant falls outside Theorem 2.3's coverage and is reported as an empirical extension; the pricing and energy diffusion configurations

in Appendix I.2 use $\alpha_{\max} = \alpha_{\min} = 0$, and the top-$k$ and shortest-path configurations keep a score function component.

Plug-in/surrogate-gradient variants for generative cost models. Both generative parameterizations admit the same plug-in/surrogate-gradient construction. At iteration t, sample K cost scenarios from the policy,

$$\hat{c}_t^{(1)}, \dots, \hat{c}_t^{(K)} \sim p_{\theta_t}(\cdot \mid x_t), \qquad \bar{c}_t := \frac{1}{K} \sum_{k=1}^{K} \hat{c}_t^{(k)},$$

and let $g_{\text{loss}}\big(\hat{c}, f_\phi(x)\big)$ denote the analytical plug-in/surrogate gradient at predicted cost $\hat{c}$ and nuisance estimate $f_\phi(x)$. We use two variants, defined as surrogate losses whose gradient with respect to θ flows only through the reparameterized samples $\hat{c}_t^{(k)}(\theta)$ and $\bar{c}_t(\theta)$, with g_{loss} evaluated at the current iterate and held fixed. The gradient of either loss is the plug-in term of the generative hybrid update below.

Per-scenario gradient variant. Evaluate the surrogate gradient separately for each generated scenario and average,

$$g_{t,k} = g_{\text{loss}}\big(\hat{c}_t^{(k)},\, f_{\phi_t}(x_t)\big), \qquad L_t^{\text{ps}}(\theta) = \frac{1}{K} \sum_{k=1}^{K} \big\langle \hat{c}_t^{(k)}(\theta),\, g_{t,k} \big\rangle.$$

For oracle-call point surrogates, this requires K surrogate evaluations.

Sample-average approximation (SAA) variant. Average scenarios first, then evaluate the surrogate gradient once at the empirical mean,

$$g_t^{\text{SAA}} = g_{\text{loss}}\big(\bar{c}_t,\, f_{\phi_t}(x_t)\big), \qquad L_t^{\text{SAA}}(\theta) = \big\langle \bar{c}_t(\theta),\, g_t^{\text{SAA}} \big\rangle.$$

For oracle-call point surrogates, SAA requires a single surrogate evaluation rather than K.

The two variants give the same gradient only when $g_{\text{loss}}(\bar{c}_t,\, f_{\phi_t}(x_t)) = \frac{1}{K} \sum_{k=1}^{K} g_{t,k}$, for example when the surrogate gradient is affine in the predicted cost or independent of it after fixing the candidate pool. For oracle-call losses such as SPO+, DBB, PG, implicit maximum likelihood estimation (IMLE), adaptive IMLE (AIMLE), and PFYL, this need not hold because the oracle solution can change with the predicted cost.

Hybrid policy update for generative cost models. Both generative models combine three gradients: the score function gradient g_t^{score} above ($g_{t,\text{CNF}}^{\text{score}}$ or $g_{t,\text{diff}}^{\text{score}}$), the plug-in term $g_t^{\text{aux}} := \nabla_\theta L_t^{\text{aux}}(\theta_t)$ with $L_t^{\text{aux}} \in \{L_t^{\text{ps}}, L_t^{\text{SAA}}\}$, and a regularizer g_t^{reg} that fits the model to the nuisance estimate $\check{c}_t := f_{\phi_t}(x_t)$. For the CNF, $g_t^{\text{reg}} = -\nabla_\theta \log p_{\theta_t}(\check{c}_t \mid x_t)$. For the diffusion model, it is the denoising mean-squared-error (MSE) regularizer

$$g_t^{\text{reg}} = \nabla_\theta\, \mathbb{E}_{\tau,\,\varepsilon}\Big[\big\|\varepsilon - \varepsilon_{\theta_t}(\check{c}_t^{(\tau)}, \tau, x_t)\big\|^2\Big],$$

where $\varepsilon \sim \mathcal{N}(0, I_{d_w})$ is the forward-process noise, $\check{c}_t^{(\tau)}$ is the forward-noised version of $\check{c}_t$ at diffusion step τ, and $\varepsilon_\theta(\cdot, \tau, x)$ is the denoiser network, which predicts that noise. The hybrid policy update is

$$g_t^{\alpha_t} = \alpha_t\, g_t^{\text{score}} + (1 - \alpha_t)\Big[\kappa_{\text{dfl}}\, g_t^{\text{aux}} + \lambda_{\text{reg}}\, g_t^{\text{reg}}\Big].$$

The hyperparameters κ_{dfl} and λ_{reg} are tuned per setting (Appendix I.2).

The $\lambda_{\text{reg}}\, g_t^{\text{reg}}$ term fits the model to the nuisance estimate $f_{\phi_t}(x_t)$. Without it, the auxiliary surrogate alone can move the sampler off the estimated conditional cost density.

H BASELINE ALGORITHMS

The contextual-bandit baselines below use the same pure bandit feedback $v_t = c_t^\top w_t$ as Algorithm 1. GreedyCB exploits a point cost predictor f_θ, ϵ-GreedyCB replaces that point prediction by a standard-normal random cost vector on exploration iterations (default $\epsilon = 0.1$), and TSCB samples from the learned conditional cost distribution.

Algorithm 2: Greedy contextual bandit (GREEDYCB)

Input: Initial parameters $\theta_1 \in \mathbb{R}^{d_\theta}$; stepsize $\eta > 0$
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t;
  Commit to $w_t \in \arg\min_{w \in \mathcal{W}} f_{\theta_t}(x_t)^\top w$;
  Observe feedback v_t;
  $L_t^{\mathrm{G}}(\theta) \leftarrow \big(f_\theta(x_t)^\top w_t - v_t\big)^2$;
  $\theta_{t+1} \leftarrow \theta_t - \eta \, \nabla_\theta L_t^{\mathrm{G}}(\theta)\big|_{\theta=\theta_t}$;
end

Algorithm 3: ϵ-Greedy contextual bandit (ϵ-GREEDYCB)

Input: Initial parameters $\theta_1 \in \mathbb{R}^{d_\theta}$; stepsize $\eta > 0$; exploration probability $\epsilon \in [0, 1]$
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t;
  Draw $U_t \sim \mathrm{Unif}[0, 1]$;
  if $U_t \le \epsilon$ **then**
    Sample a random exploratory cost vector $\tilde{c}_t \sim \mathcal{N}(0, I_{d_w})$;
  else
    $\tilde{c}_t \leftarrow f_{\theta_t}(x_t)$;
  end
  Commit to $w_t \in \arg\min_{w \in \mathcal{W}} \tilde{c}_t^\top w$;
  Observe feedback v_t;
  $L_t^{\epsilon\text{-G}}(\theta) \leftarrow \big(f_\theta(x_t)^\top w_t - v_t\big)^2$;
  $\theta_{t+1} \leftarrow \theta_t - \eta \, \nabla_\theta L_t^{\epsilon\text{-G}}(\theta)\big|_{\theta=\theta_t}$;
end

Algorithm 4: Thompson-style contextual bandit (TSCB)

Input: Initial parameters $\theta_1 \in \mathbb{R}^{d_\theta}$ of a conditional distribution $p_\theta(\cdot \mid x)$; stepsize $\eta > 0$
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t;
  Sample $\hat{c}_t \sim p_{\theta_t}(\cdot \mid x_t)$;
  Commit to $w_t \in \arg\min_{w \in \mathcal{W}} \hat{c}_t^\top w$;
  Observe feedback v_t;
  Let $q_\theta(\cdot \mid x_t, w_t)$ be the distribution of $c^\top w_t$ induced by $c \sim p_\theta(\cdot \mid x_t)$;
  $L_t^{\mathrm{TS}}(\theta) \leftarrow -\log q_\theta(v_t \mid x_t, w_t)$;
  $\theta_{t+1} \leftarrow \theta_t - \eta \, \nabla_\theta L_t^{\mathrm{TS}}(\theta_t)$;
end

Semi-bandit adaptation. Under semi-bandit feedback, the CB baselines retain their action-selection rules but replace the scalar squared-loss update by the vector form $\|v_t - H(w_t)\, f_\theta(x_t)\|^2$, i.e., the same nuisance least-squares objective minimized by f_ϕ in Algorithm 1, applied to the CB cost predictor. For coordinate-mask observations $H(w_t) = \mathrm{diag}(w_t)$ this reduces to a per-coordinate masked regression on the action-selected entries. Under semi-bandit feedback the CB baselines keep their pure-bandit learning rates, so the comparison isolates the richer feedback signal.

Linear bandit baselines (OFUL and LinTS). Both algorithms use the same linear cost model f_θ as the Gaussian linear variant of the other methods. Because this model is linear in θ, the predicted cost of decision w in context x is an inner product, $f_\theta(x)^\top w = \theta^\top z(x, w)$, where the feature vector $z(x, w)$ combines the decision and the context. For example, on top-k, shortest path, and pricing the model is $f_\theta(x) = \Theta x$, where $\Theta \in \mathbb{R}^{d_w \times d_x}$ is a coefficient matrix whose entries form θ. The predicted cost is then $w^\top \Theta x = \sum_{i,j} \Theta_{ij}\, w_i x_j$, so $z(x, w)$ collects the products $w_i x_j$. OFUL and LinTS treat $z(x, w)$ as the feature vector of decision w. Both keep the ridge statistics V_t and b_t of the observed

pairs (z_s, v_s), with $z_s = z(x_s, w_s)$, and use the noise scale $R = 1$. We write $\|u\|_A = \sqrt{u^\top A u}$. OFUL (Algorithm 5) minimizes a lower confidence bound on the cost, with the confidence radius ρ_t computed from Abbasi-Yadkori et al. (2011, Theorem 2) at $\delta = 0.05$ and parameter-norm bound $S = 5$. Its bonus is nonlinear in w, so each round enumerates $\mathcal{W}$ (105 decisions on top-k, 70 paths on shortest path). LinTS (Algorithm 6) samples a parameter from the Gaussian posterior, which starts from the prior $\mathcal{N}(0, R^2 I)$, and calls the linear oracle once per round.

Algorithm 5: Optimism in the face of uncertainty for linear bandits (OFUL)

Input: Feature map z; noise scale R; confidence level $\delta \in (0, 1)$; parameter-norm bound S
$V_1 \leftarrow I$, $b_1 \leftarrow 0$;
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t;
  $\hat{\theta}_t \leftarrow V_t^{-1} b_t$ and $\rho_t \leftarrow R\sqrt{\log\det V_t + 2\log(1/\delta)} + S$;
  Commit to $w_t \in \arg\min_{w \in \mathcal{W}} \{\hat{\theta}_t^\top z(x_t, w) - \rho_t \|z(x_t, w)\|_{V_t^{-1}}\}$, by enumerating $\mathcal{W}$;
  Observe feedback v_t and set $z_t \leftarrow z(x_t, w_t)$;
  $V_{t+1} \leftarrow V_t + z_t z_t^\top$ and $b_{t+1} \leftarrow b_t + v_t z_t$;
end

Algorithm 6: Linear Thompson sampling (LinTS)

Input: Feature map z and linear cost model f_θ; noise scale R
$V_1 \leftarrow I$, $b_1 \leftarrow 0$;
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t;
  $\hat{\theta}_t \leftarrow V_t^{-1} b_t$;
  Sample $\tilde{\theta}_t \sim \mathcal{N}(\hat{\theta}_t,\, R^2 V_t^{-1})$;
  Commit to $w_t \in \arg\min_{w \in \mathcal{W}} f_{\tilde{\theta}_t}(x_t)^\top w$;
  Observe feedback v_t and set $z_t \leftarrow z(x_t, w_t)$;
  $V_{t+1} \leftarrow V_t + z_t z_t^\top$ and $b_{t+1} \leftarrow b_t + v_t z_t$;
end

I Hyperparameter selection and compute

I.1 Tuning protocol

Hyperparameters were selected by random search per (benchmark, distributional cost model) on dedicated validation runs that share no seeds with the reported paper runs. Validation used 10 seeds for top-k and shortest path, and 5 seeds for pricing and energy; the paper results use 30 seeds. Validation horizons were $T = 2{,}000$ for the point-model sweep, $T = 5{,}000$ for a two-stage generative screen, and $T = 394$ on the energy `load3` split (validation fraction 0.5); paper horizons are $T = 2{,}000$ for point models ($T = 10{,}000$ for pricing, whose raw-context linear model has 2,000 parameters: at $T = 2{,}000$ every method on pricing is still within 11% of the random policy, and by $T = 10{,}000$ all methods have learned enough to separate from it), $T = 15{,}000$ for the generative comparison, and the held-out experiment split for energy. Candidates were ranked by mean final cumulative regret on the validation seeds. For the pricing generative models, the $T = 5{,}000$ screen covered 48 CNF and 72 diffusion candidates, and the top three per model were confirmed on 10 validation seeds. Full search spaces and per-benchmark sweep specifications are available in the supplementary code.

I.2 Selected configurations

Across (benchmark, distributional cost model) settings, the selected configurations share the following defaults:

- Policy and nuisance learning rates on the same shifted-inverse schedule, $\eta^t = \eta^0/(1 + t/o)$ with offset $o = 100$. The reported η_θ^0 and η_ϕ^0 are the initial values of the policy step size (β in Algorithm 1) and the nuisance step size (η_t).

- Adaptive mixing-weight schedule (Appendix G.2): smoothing factor 0.05 on α_t updates, fixed-warmup gate of $T_{\text{warm}} = 100$ iterations.
- For pricing, energy, top-k NN, and the generative models, the score function and plug-in gradients are L_2-normalized separately before the convex mix $\alpha\, g_t^{\text{score}} + (1-\alpha)\, g_t^{\text{plug-in}}$, so that α sets the direction independently of the raw magnitudes. The top-k Gaussian linear and shortest-path Gaussian configurations use raw gradients.

Surrogate names follow Appendix G.3; baselines are moving-average (MA) and nuisance-induced (NI) from Appendix G.1.

Table 2 reports the point-model configurations across all four benchmarks. Table 3 reports the generative-model configurations across the same four benchmarks. All configurations use the adaptive mixing-weight schedule of Appendix G.2. In the energy configurations and the pricing CNF configuration, the mixing weight decays over a time scale given in column τ_{sched}. The other configurations hold it at $\alpha_{\max}$ for the first 100 iterations and then switch to $\alpha_{\min}$, so their τ_{sched} entry is –. Top-k and shortest-path entries use $\deg = 8$ and pricing entries $\deg = 6$; synthetic distributional-model comparisons use $T = 15{,}000$, and energy uses its native test-window length.

Table 2: Point-model configurations across the four benchmarks. Columns: initial policy learning rate (LR) η_θ^0, initial nuisance LR η_ϕ^0, plug-in surrogate, exploration scale σ_{G}, mixing bounds $\alpha_{\max}, \alpha_{\min}$, time scale τ_{sched} over which the mixing weight decays (– for configurations that hold $\alpha_{\max}$ for the first 100 iterations instead, Appendix G.2), baseline.

Benchmark	Class	η_θ^0	η_ϕ^0	Surrogate	σ_{G}	$\alpha_{\max}$	$\alpha_{\min}$	τ_{sched}	Baseline
Top-k	Gaussian linear	6.8e-2	5.3e-2	pairwiseDiff	8.6e-1	0.3	0.02	–	MA
Top-k	Gaussian NN	7.5e-2	1.1e-2	IMLE	2.6e-1	0.1	0.05	–	MA
Shortest path	Gaussian linear	3.0e-2	1.5e-2	pairwiseDiff	3.9e-1	0.5	0.05	–	NI
Shortest path	Gaussian NN	3.0e-2	1.0e-2	pairwiseDiff	5.3e-1	0.3	0.02	–	NI
Pricing	Gaussian linear	8.2e-1	1.6e-3	DBB	4.9e-5	0.5	0.05	–	NI
Pricing	Gaussian NN	5.3e-1	1.0e-3	DBB	1.4e-5	0.5	0.05	–	NI
Energy	Gaussian linear	5.0e-3	2.5e-2	MAP-C	1.9e-3	0.1	0.0	100	NI
Energy	Gaussian NN	1.0e-3	1.0e-3	DBB	5.0e-3	0.8	0.2	100	MA

Table 3: Generative-model configurations across the four benchmarks. The auxiliary plug-in path uses the per-scenario gradient (Appendix G.4) with the auxiliary surrogate as listed, K scenarios per iteration, surrogate-loss weight κ_{dfl}, and generative-regularizer weight λ_{reg}. Energy CNF uses two coupling layers with hidden width 128; energy diffusion uses 20 training and 8 inference timesteps with ELBO weighting.

Benchmark	Class	η_θ^0	η_ϕ^0	$\alpha_{\max}$	$\alpha_{\min}$	τ_{sched}	Baseline	Aux. surrogate	K	κ_{dfl}	λ_{reg}
Top-k	CNF	3.0e-2	6.0e-3	0.2	0.01	–	NI	pairwiseDiff	32	3.0	1.0
Top-k	Diffusion	1.3e-2	1.3e-2	0.2	0.01	–	MA	pairwiseLTR	200	10.0	1.0
Shortest path	CNF	9.0e-3	6.8e-3	0.5	0.02	–	MA	SPO+	128	1.0	0.03
Shortest path	Diffusion	1.0e-2	1.0e-2	0.1	0.0	–	NI	pairwiseDiff	200	1.0	0.1
Pricing	CNF	9.0e-2	9.0e-2	0.5	0.02	500	NI	pairwiseLTR	200	10.0	0.3
Pricing	Diffusion	7.4e-3	1.0e-3	0.0	0.0	–	MA	pairwiseLTR	64	3.0	0.3
Energy	CNF	1.5e-2	1.1e-2	0.8	0.05	500	NI	pairwiseLTR	100	1.0	0.1
Energy	Diffusion	3.0e-3	3.0e-3	0.0	0.0	100	NI	pairwiseLTR	100	1.0	0.1

The pricing-diffusion and energy-diffusion entries use $\alpha_{\max} = \alpha_{\min} = 0$ (plug-in only), consistent with Appendix J.5.

I.3 Compute resources

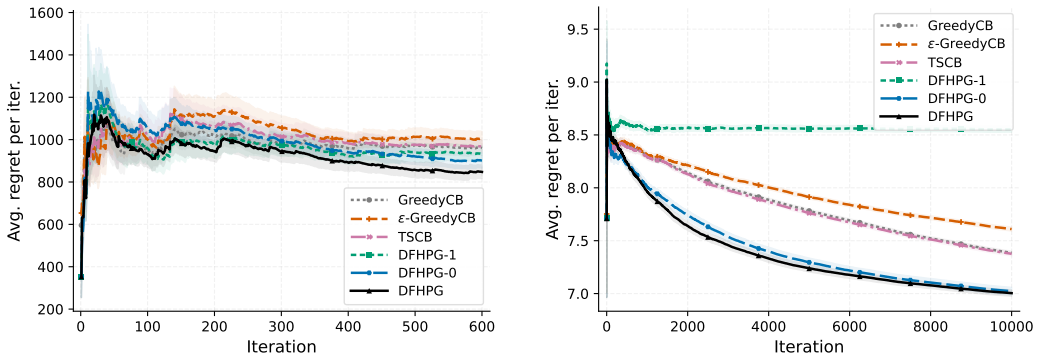
All experiments ran on a shared high-performance computing cluster running Red Hat Enterprise Linux 9.6. Each CPU trial uses one Intel Xeon Gold 6226 core at 2.70 GHz with 2 GB of RAM. GPU trials use a single NVIDIA L40S or RTX 6000.

J ADDITIONAL EXPERIMENTAL RESULTS

This section collects supplementary tables and figures for the numerical study. The main text reports the headline comparisons; the appendix gives convergence plots, sensitivity analyses, and the feedback-mode comparison. Compute resources used to run the experiments are reported in Appendix I.3.

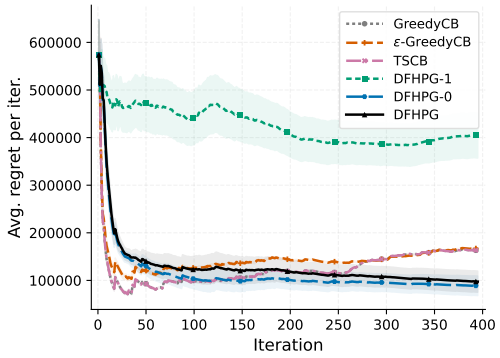
J.1 CONVERGENCE PLOTS

The main text reports the top-k convergence plot. Figure 3 gives the corresponding shortest-path, pricing, and real-data energy-scheduling trajectories, zoomed into the first 600 iterations of the $T = 2{,}000$ run for shortest path and over the full $T = 10{,}000$ run for pricing. On shortest path, all methods pay for the noisy first iterations, but the hybrid curve settles below the contextual-bandit baselines and remains the lowest curve through the later part of this window. The pricing panel shows the full $T = 10{,}000$ horizon: DFHPG and DFHPG-0 separate from the contextual-bandit baselines after the first few hundred iterations and stay lowest, while DFHPG-1 stays highest.



(a) Shortest path

(b) Pricing



(c) Energy scheduling

Figure 3: Convergence plots for shortest path, pricing, and energy scheduling. Shortest path is zoomed to the first 600 iterations, pricing shows its full $T = 10{,}000$ horizon, and energy scheduling uses its native test-window length. Each curve reports average regret per iteration, with shaded ± 1 standard-error bands across replications.

J.2 STATIONARITY MEASUREMENT

On top-k and shortest path the conditional mean $f^\star$ is known to the evaluator, so $\nabla_\theta V^{\mathrm{cost}}(\theta) = \mathbb{E}\big[f^\star(x)^\top w^\star(\hat{c})\, \nabla_\theta \log p_\theta(\hat{c} \mid x)\big]$ can be estimated without differentiating through the oracle. At every iterate we form two independent Monte Carlo batch means of this expression, each from 2,000

samples, and take their inner product, which is an unbiased estimate of $\|\nabla_\theta V^{\mathrm{cost}}(\theta_t)\|^2$; negative estimates are kept. The runs use the fixed-α configuration at $\alpha = 0$, where the nuisance-bias term of Theorem 2.2 is largest (constant policy and nuisance step sizes, raw gradients without normalization) with the Table 1 replications, the surrogate of Table 2, and the neural-network nuisance model. The fixed-α ablation uses the same configuration with $\alpha \in \{0, 0.25\}$: $\alpha = 0.25$ lowers cumulative expected cost on top-k from 3.03×10^5 to 2.31×10^5 (better in 25 of 30 replications) and performs comparably to $\alpha = 0$ on shortest path (better in 14 of 30). Pricing is excluded: its selected sampling scale $\sigma_{\mathrm{G}} \approx 5 \times 10^{-5}$ makes the score-function estimate too noisy at feasible sample sizes.

Table 4: Ergodic averages up to iteration t of $\|\nabla_\theta V^{\mathrm{cost}}(\theta_s)\|^2$ and of per-round regret, fixed-α configuration at $\alpha = 0$. Mean $\pm$ standard error over 30 replications.

t	100	250	500	1000	2000
Top-k					
$\|\nabla V^{\mathrm{cost}}\|^2$ ($\times10^5$)	1.50 ± 0.13	0.98 ± 0.09	0.66 ± 0.06	0.44 ± 0.04	0.35 ± 0.05
Regret per round	310 ± 38	277 ± 21	224 ± 17	170 ± 10	148 ± 7
Shortest path					
$\|\nabla V^{\mathrm{cost}}\|^2$ ($\times10^5$)	8.68 ± 1.86	4.35 ± 0.91	1.83 ± 0.60	1.40 ± 0.41	1.02 ± 0.23
Regret per round	1009 ± 77	996 ± 54	904 ± 36	876 ± 36	762 ± 27

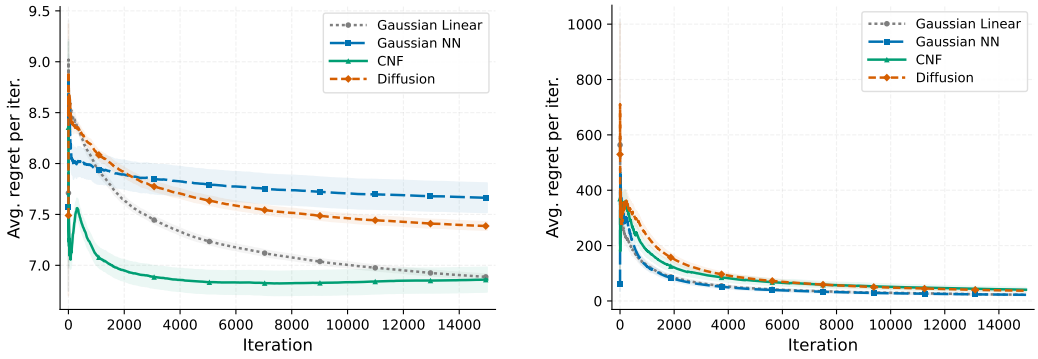
J.3 Distributional cost models

Figure 2 compares distributional cost models on pricing at the extended horizon $T = 15{,}000$. CNF has the lowest average regret for most of the run but levels off after about $5{,}000$ iterations, while the Gaussian linear model keeps improving and nearly matches it at $T = 15{,}000$ (6.86 ± 0.12 and 6.89 ± 0.03 per iteration). The Gaussian NN and diffusion models remain above both (7.66 ± 0.15 and 7.39 ± 0.04). Figure 4 reports the same comparison on top-k selection, shortest path, and energy scheduling; the synthetic benchmarks use $T = 15{,}000$ and energy uses its native test-window length. On top-k selection the Gaussian models finish slightly below CNF and diffusion, and on shortest path the Gaussian linear model has the lowest final regret. On energy scheduling the Gaussian NN has substantially higher regret than the other three models. No single cost model dominates across benchmarks, and the DFHPG framework accommodates all four.

J.4 Alternative surrogate choices

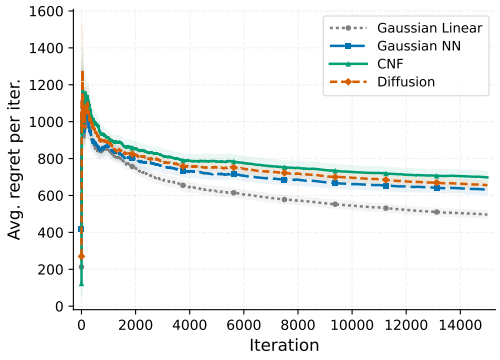
The surrogate loss for the decision-focused plug-in component is treated as a tunable hyperparameter rather than fixed to a default. This appendix reports the per-benchmark sweep over standard surrogate choices, implemented following the PyEPO reference implementations (Tang & Khalil, 2024), across the pricing, shortest-path, top-k, and energy-scheduling benchmarks. All synthetic comparisons in this section use the Gaussian linear cost model; the energy comparison uses the same setup as in the main results. Each heatmap changes only the surrogate loss used by the decision-focused component; the score function update, nuisance update, contexts, feasible set, and feedback model are held fixed. The two columns correspond to the full hybrid update DFHPG and the plug-in-only update DFHPG-0. Cell values are mean final cumulative regret, so lower values are better.

Figure 5 shows that no single surrogate dominates across benchmarks, motivating the per-problem tuning protocol. On pricing, listwiseLTR, MSE+SPOPlus, IMLE, and SPOPlus (tied with SPOCaching) lead in both columns, while PG, NID, and DPO trail, and pointwiseLTR also trails for DFHPG. On shortest path, DFHPG is approximately flat across a wide best group (AIMLE, pairwiseDiff, NID, DPO, PFYL, PG), whereas DFHPG-0 is more variable and peaks at SPOPlus, SPOCaching, and AIMLE. On top-k selection, NID, pairwiseDiff, and MSE+SPOPlus lead for DFHPG, while DFHPG-0 prefers NID and SPO-type losses. On energy scheduling the ranking inverts relative to pricing: pairwiseDiff, MSE+SPOPlus, IMLE, and NCE lead, while DBB and contrastiveMAP fall to the bottom half, and PG and DPO collapse by an order of magnitude.

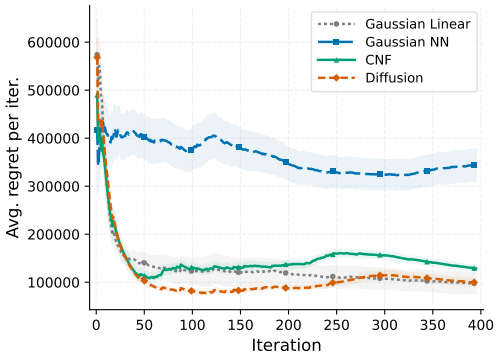


(a) Pricing ($T = 15{,}000$)

(b) Top-k selection ($T = 15{,}000$)



(c) Shortest path ($T = 15{,}000$)



(d) Energy scheduling

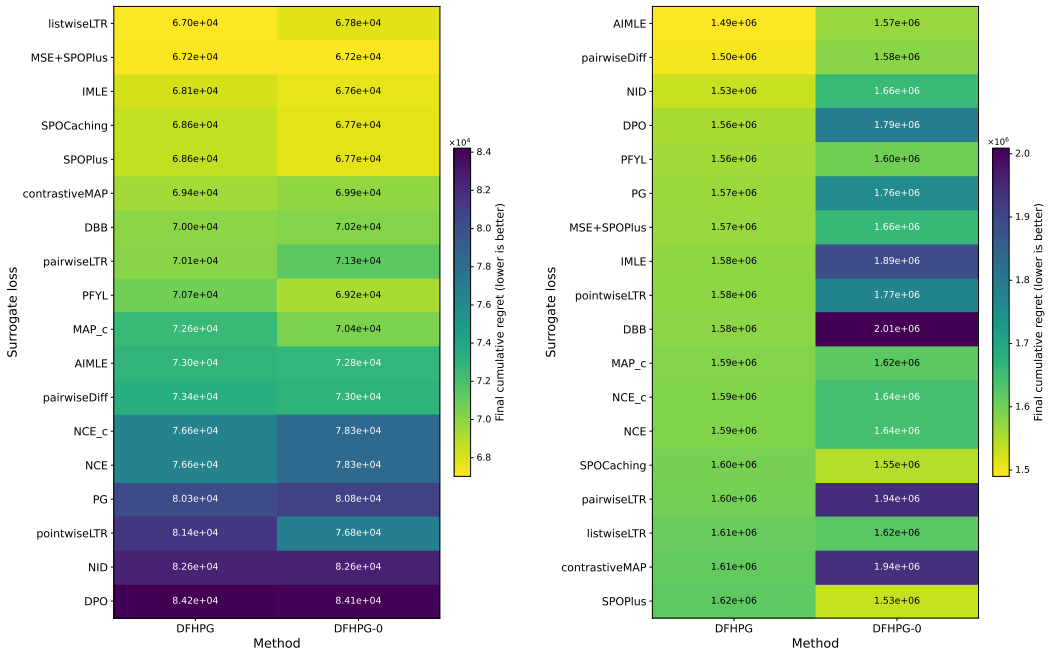
Figure 4: Comparison of distributional cost models on pricing, top-k selection, shortest path, and energy scheduling. Each curve reports average regret per iteration, with shaded ± 1 standard-error bands across replications.

J.5 Mixing-weight sensitivity

Figure 6 ablates the mixing weight between the score function and plug-in components. We compare the adaptive schedule from Appendix G.2 against fixed values of α on all four benchmarks. The adaptive schedule is a strong default but does not strictly dominate the best fixed α: on top-k selection, shortest path, and pricing it matches or modestly improves on the strongest fixed setting; on the energy-scheduling benchmark it trails the smaller fixed α values.

J.6 Sensitivity across problem difficulty

Figure 7 reports degree-robustness on pricing, shortest path, and top-k selection, where the polynomial degree controls the nonlinearity of the synthetic cost generator (top-k and shortest path) or the demand generator (pricing). On top-k and shortest path, higher degrees act as a difficulty knob: every method's regret rises with degree under the Gaussian linear cost model. On pricing, the degree changes the shape of the willingness to pay while its typical size stays fixed (Appendix F.3), so regret varies little with degree. Across degrees, DFHPG and DFHPG-0 retain a regret advantage over the contextual-bandit baselines, but the magnitude of the gap is benchmark-specific. On pricing, where DFHPG is tuned separately at each degree, the better of DFHPG and DFHPG-0 is below the best contextual-bandit baseline at every degree, by 1–10%. On top-k the gap is more modest at low degrees (around $2\times$ at $\deg = 2$) and widens to roughly 4–$5\times$ at $\deg \geq 4$. Shortest path shows small gaps throughout. The score-only DFHPG-1 ablation tracks or trails the baselines, indicating that the decision-focused plug-in component is what drives the advantage as the problem hardens.



(a) Pricing

(b) Shortest path

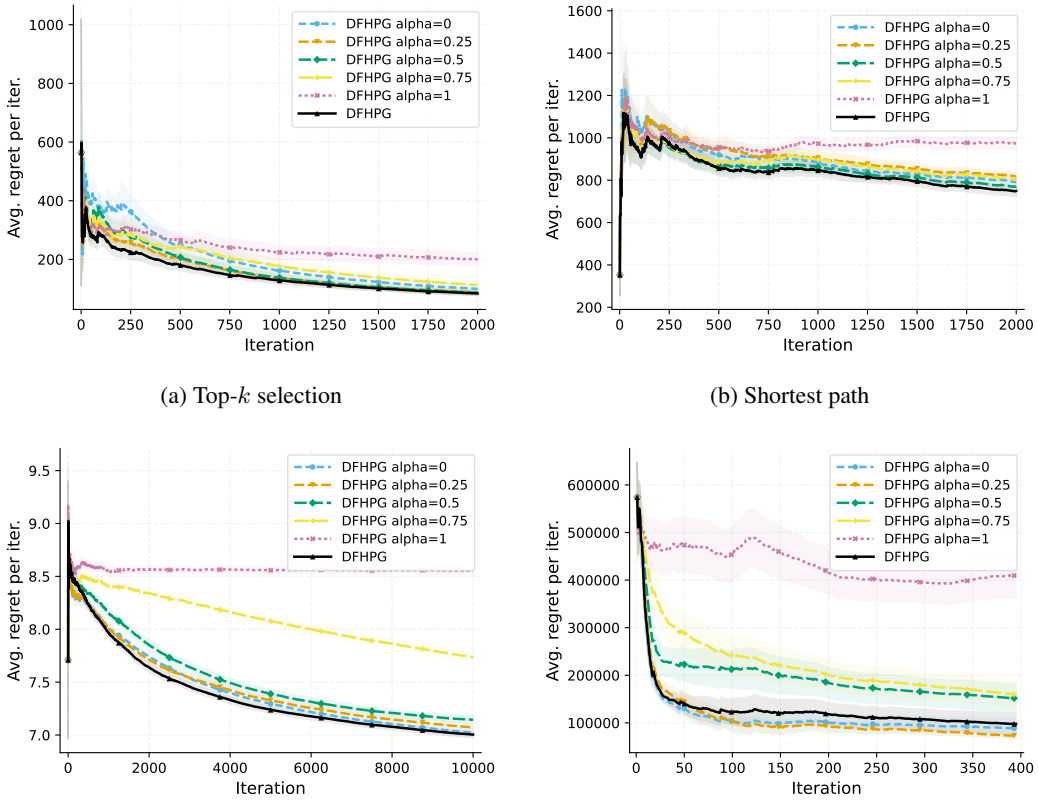
(c) Top-k selection

(d) Energy scheduling

Figure 5: Surrogate-loss comparison across the four benchmark problems. Each heatmap reports mean final cumulative regret for DFHPG and DFHPG-0 when changing only the surrogate loss used by the decision-focused component; lower values are better.

J.7 Feedback mode comparison

Figure 8 compares final cumulative regret across three feedback modes (pure bandit, semi-bandit, and full information) for both DFHPG and DFHPG-0 on each benchmark. On all four problems,



(a) Top-k selection

(b) Shortest path

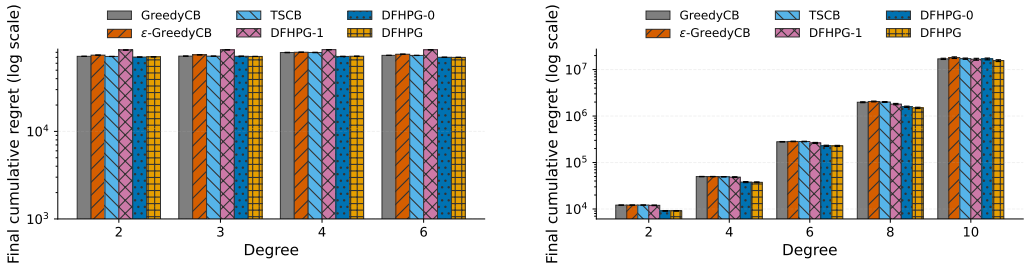
(c) Pricing

(d) Energy scheduling

Figure 6: Mixing-weight ablation comparing the adaptive α_t schedule against fixed α on the four benchmarks. Each curve reports average regret per iteration, with shaded ± 1 standard-error bands across replications.

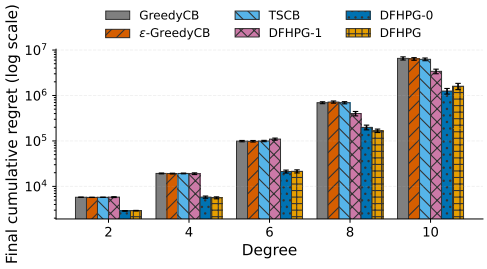
semi-bandit and full-information feedback lower regret relative to pure bandit feedback. Semi-bandit feedback sharpens the nuisance fit, whereas under full information DFHPG uses the single realized cost vector as its plug-in target (Appendix F). On pricing, where Poisson demand makes that target noisy, full information is about 20% above semi-bandit feedback, although still below pure bandit feedback. On the energy benchmark the variance contraction is especially large, with pure-bandit error bars roughly an order of magnitude wider than under semi-bandit or full feedback.

Table 5 reports final cumulative regret under semi-bandit feedback for the contextual-bandit baselines and DFHPG variants. The CB baselines use the semi-bandit-adapted regression update described at the end of Appendix H; their action-selection rules are unchanged. DFHPG-0 is best on top-k, shortest path, and energy. On pricing, the semi-bandit-adapted TSCB and GREEDYCB are lowest, about 13% below DFHPG-0. Under semi-bandit feedback, each round reveals the revenue of every product at its chosen price, so the regression these baselines run has a direct target for 20 of the 80 cost entries per round, and solving with the fitted costs already prices well. DFHPG-0 learns from the same observations only through a separate nuisance model and a surrogate update of the policy, an indirect path designed for scarce, aggregate feedback.



(a) Pricing

(b) Shortest path

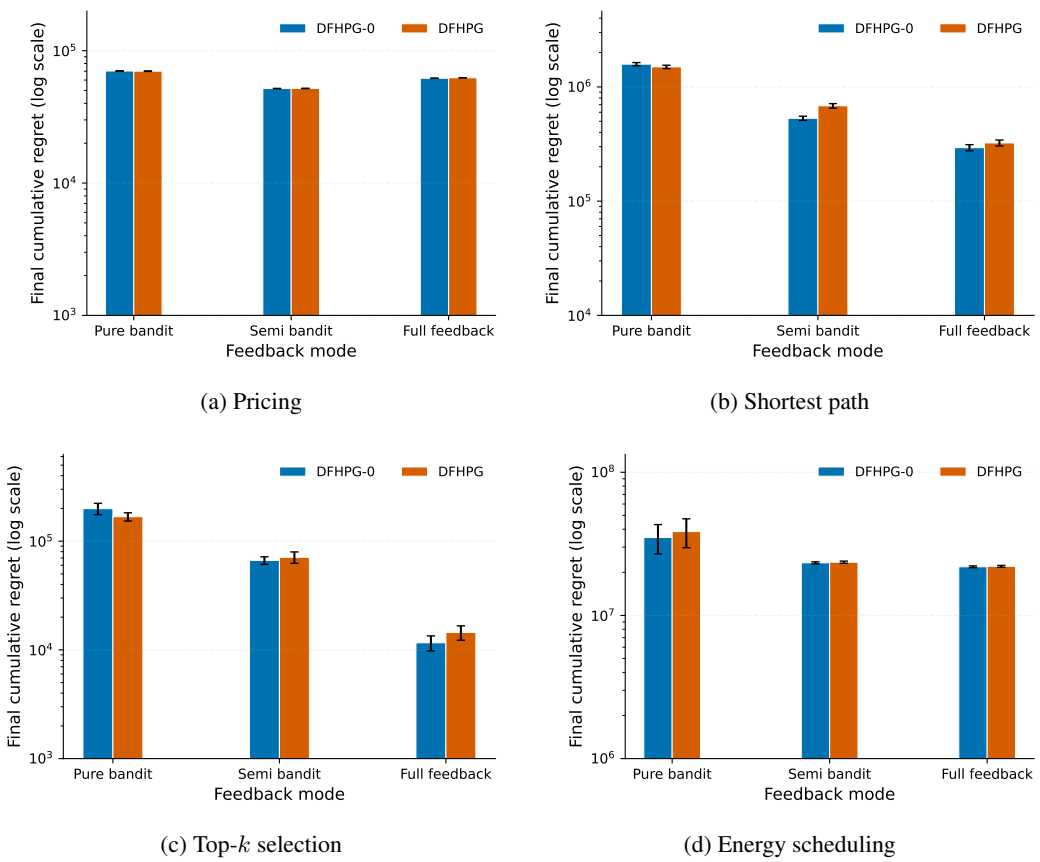


(c) Top-k selection

Figure 7: Sensitivity across problem difficulty for pricing, shortest path, and top-k selection. Each plot compares final cumulative regret across polynomial degrees for the Gaussian linear cost-model experiments. Lower is better.

Table 5: Final cumulative regret under semi-bandit feedback. The contextual-bandit baselines use the semi-bandit-adapted regression update of Appendix H. Setup matches Table 1; the best result in each column is bolded.

Method	Top-k (deg. 8)	Shortest path (deg. 8)	Pricing (deg. 6)	Energy
GreedyCB	$3.01\times10^{5} \pm 1.61\times10^{4}$	$9.37\times10^{5} \pm 2.32\times10^{4}$	$4.49\times10^{4} \pm 1.5\times10^{2}$	$2.66\times10^{7} \pm 1.07\times10^{5}$
ϵ-GreedyCB	$2.91\times10^{5} \pm 2.01\times10^{4}$	$9.87\times10^{5} \pm 3.29\times10^{4}$	$4.85\times10^{4} \pm 1.2\times10^{2}$	$4.16\times10^{7} \pm 4.83\times10^{5}$
TSCB	$2.87\times10^{5} \pm 1.86\times10^{4}$	$9.67\times10^{5} \pm 3.31\times10^{4}$	$\mathbf{4.49\times10^{4} \pm 1.5\times10^{2}}$	$2.66\times10^{7} \pm 1.08\times10^{5}$
DFHPG	$7.10\times10^{4} \pm 8.52\times10^{3}$	$6.83\times10^{5} \pm 3.11\times10^{4}$	$5.19\times10^{4} \pm 1.0\times10^{2}$	$2.35\times10^{7} \pm 4.21\times10^{5}$
DFHPG-0	$\mathbf{6.65\times10^{4} \pm 5.30\times10^{3}}$	$\mathbf{5.31\times10^{5} \pm 2.28\times10^{4}}$	$5.18\times10^{4} \pm 9.8\times10^{1}$	$\mathbf{2.33\times10^{7} \pm 3.78\times10^{5}}$



(a) Pricing

(b) Shortest path

(c) Top-k selection

(d) Energy scheduling

Figure 8: Feedback mode comparison across the four benchmarks. Each plot compares final cumulative regret across feedback modes (bandit, semi-bandit, full information) on a log-scaled regret axis. Synthetic panels use the Gaussian linear cost-model experiments. Lower is better.