Received: 22 May 2020 | Accepted: 19 January 2022

DOI: 10.1111/poms.13683

ORIGINAL ARTICLE

Data-driven storage operations: Cross-commodity backtest and structured policies

Christian Mandl[1] | Selvaprabu Nadarajah[2] | Stefan Minner[3] | Srinagesh Gavirneni[4]

[1] School of Management, Deggendorf Institute of Technology, Deggendorf, Germany

[2] College of Business Administration, University of Illinois at Chicago, Chicago, Illinois, USA

[3] TUM School of Management, Technische Universität München, München, Germany

[4] Samuel Curtis Johnson Graduate School of Management, Cornell University, Ithaca, New York, USA

Correspondence
Christian Mandl, School of Management, Deggendorf Institute of Technology, Deggendorf, Bayern, Germany.
Email: christian.mandl@th-deg.de

Handling Editor: Sridhar Seshadri

Abstract

Storage assets are critical for physical trading of commodities under volatile prices. State-of-the-art methods for managing storage facilities such as the reoptimization heuristic (RH), which are part of commercial software, approximate a Markov Decision Process (MDP) assuming full information regarding the state and the stochastic commodity price process and hence suffer from informational inconsistencies with observed price data and structural inconsistencies with the true optimal policy, which are both components of generalization error. Focusing on spot trades, we find via an extensive backtest that this error can lead to significantly suboptimal RH policies. We develop a forward-looking data-driven approach (DDA) to learn policies and reduce generalization error. This approach extends standard (backward-looking) DDA in two ways: (i) It represents historical and estimated future profits as functions of features in the training objective, which typically includes only past profits; and (ii) it enforces structural properties of the optimal policy. To elaborate, DDA trains parameters of bang-bang and base-stock policies, respectively, using linear- and mixed-integer programs, thereby extending known DDAs that parameterize decisions as functions of features without policy structure. We backtest the performance of RH and DDA on six major commodities, employing feature selection across data from Reuters, Bloomberg, and other public data sets. DDA can improve RH on real data, with policy structure needed to realize this improvement. Our research advances the state-of-the-art for storage operations and can be extended beyond spot trading to handle generalization error when also including forward trades.

KEYWORDS

commodity storage, data-driven optimization, generalization error, price uncertainty, reoptimization

1 | INTRODUCTION

Storage assets play a fundamental role in commodity markets. Examples include natural gas and oil storage caverns, grain silos, and metal warehouses. In 2017, more than 600 warehouses across 14 countries were approved by the London Metal Exchange (LME), with several million tonnes of metal delivered into and taken out of LME warehouses (LME, 2017). The Chicago Merchantile Exchange (CME), another major marketplace for physical metal trading, is also expanding its storage network. CME-registered warehouses in Salt Lake City held 131,774 tonnes of copper in January 2018 (Reuters, 2018), which translated into 927 million USD of inventory.

Merchant trading companies use storage to benefit from positive commodity price differentials over time, that is, they buy low, store, and sell high at a later date (Williams & Wright, 1991, p. 24). These assets have finite space and

Accepted by Sridhar Seshadri, after three revisions.

constraints on the rates of injection and withdrawal. Maximizing the profit from operating storage requires adapting the timing of constrained injections and withdrawals to the movement of uncertain commodity spot prices. The related optimization of storage operations can be approached using a Markov decision process (MDP) that contains in its state the on-hand inventory (i.e., endogenous state) and multiple factors (i.e., exogenous state) of a Markovian stochastic process describing the evolution of spot prices. The extant storage literature formulates this MDP assuming that the stochastic process is known. A common choice for the exogenous state is a vector of futures contract prices (Lai et al., 2010) since they are available for commodities with futures markets and expectations of the spot price equals futures prices under the risk-neutral measure.

Storage MDPs with realistic price dynamics are high-dimensional and thus intractable to solve directly. Least-squares Monte Carlo (LSM) and reoptimization heuristics (RH) are state-of-the-art approaches (Breslin et al., 2008; Breslin et al., 2009; Gray & Khandelwal, 2004a, 2004b) for approximating the aforementioned intractable MDP and are part of commercial storage management software (Energy Quants, 2018; Kyos, 2018; Lacima, 2018; MathWorks, 2018). LSM computes a parametric approximation of the MDP value function using backward induction and regression, which is then used to compute storage decisions (Nadarajah et al., 2015). RH obtains storage decisions at a given stage and state by solving a deterministic linear program, referred to as an intrinsic linear program (ILP), which is based on futures prices available at the current time. For RH, ILP is reoptimized at each stage after accounting for updated futures price information (Lai et al., 2010; Secomandi, 2015). An advantage of RH over LSM is that its inputs are agnostic to the assumed commodity price process in the MDP. Moreover, in the context of natural gas, RH storage policies have been shown to be near-optimal in computational studies that assume *full information* about the storage MDP, that is, in a setting where the exogenous state composition and the stochastic process describing the evolution of this state are known and exact (e.g., Lai et al., 2010; Nadarajah & Secomandi, 2018; Secomandi, 2010; Secomandi, 2015; Wu et al., 2012). Empirical studies investigating the performance of RH for managing the storage of commodities other than natural gas are scant.

Focusing on spot trades, we perform an extensive backtest of the RH policy on price data across six commodities (i.e., copper, gold, crude oil, natural gas, corn, soybean) from Thomson Reuters over the period 2000–2017. The goal of this backtest is to understand the true performance of RH by applying its decisions on a historical sample path of prices and benchmarking the resulting profit against the value of an optimal perfect foresight solution on this price path, which is indeed optimistic but immune to assumptions implicit in the MDP. We observe that several insights regarding the performance of RH change fundamentally as explained next:

- RH yields smaller profits than ILP on 37.0% of our commodity backtest instances, which suggests that the value of reoptimization can be negative, deviating from the substantial positive value of reoptimization reported for the full-information problem (Secomandi, 2015).
- A one-period look-ahead policy leads to higher profits than RH on several instances, that is, ignoring futures price information may be beneficial. This result differs from the literature on forecast horizons in the full-information setting, which argues that far-ahead futures price information does not affect optimal first-stage decisions (Cruise et al., 2019).
- RH yields an average value of 11.0% of the perfect foresight solution, which makes one wonder if the near-optimality of RH in the full-information setting extends to real data.

We rationalize the aforementioned stark differences using the train-test paradigm of machine learning (ML). Specifically, existing performance evaluations of RH (as well as other methods such as LSM) in the storage literature both compute policy parameters/decisions (i.e., train the policy) and test the performance of these decisions under the full-information setting, which the actual data may not satisfy. Informally speaking, the performance difference in the training environment (i.e., full-information setting) and the testing environment is referred to as generalization error. Our backtest suggests this error may be significant when employing the RH policy.

Motivated by the above observations, we take an ML approach to target the reduction of generalization error and learn storage policies. To this end, we relax the full-information assumption and formulate a feature-based storage stochastic dynamic program (F-SDP) where the exogenous state is represented by a generic set of features that evolve according to an unknown stochastic process. We then develop a data-driven approach (DDA) to tackle F-SDP that extends existing approaches in two key ways. First, it is forward-looking and uses financial-market features (e.g., futures prices) to include future estimates of profit in the training objective, in addition to historical profits considered by standard DDAs (see, e.g., Bertsimas & Kallus, 2020). Second, it allows one to enforce structural properties of an optimal F-SDP policy when computing data-driven policies. Within this framework, we begin by considering standard linear decision rules (DDA-LDRs) from the literature (see, e.g., Ban & Rudin, 2019) that specify decisions as a linear parameterization of random variables. The DDA-LDR parameters are trained using the empirical risk minimization (ERM) framework (Friedman et al., 2001), which involves solving a regularized convex program. DDA-LDRs do not encode any structure of the F-SDP optimal policy and thus training them in our forward-looking approach allows us to understand the value of such future information alone without considering the impact of policy structure. We subsequently propose structured data-driven policies (DDA-SPs) that encode bang-bang and double base-stock structures shared by the F-SDP optimal policy for storage assets with different operating characteristics. In contrast to DDA-LDRs, DDA-SPs are parameterized by coefficients of price thresholds or base-stock levels, which are trained using linear and mixed-integer programming. We discuss how the regularized training procedure and the policy structure used when computing DDA-SP make it robust to price uncertainty (i.e., it accounts for downside risk) and estimation error, respectively.

We perform a backtest of DDA approaches across the same six commodities used in our RH backtest. As candidate features, we consider spot and futures prices from Thomson Reuters, analyst forecasts of spot prices from Bloomberg, temperature, the S&P 500 index, and the Trade Weighted U.S. Dollar index. Feature selection reveals several practical insights. First, in the absence of futures prices and analyst forecasts as features, adding the S&P 500 and trade weighted U.S. dollar indices can improve profits compared to using only spot prices. This finding is relevant when futures markets or analyst forecasts are absent, which is the case for commodities such as asphalt and specific types of polyethylene. Second, while futures prices have large errors when treated as forecasts of spot prices, their inclusion on top of spot prices can improve storage profits. Third, embedding analyst forecasts further enhances storage profits by 7%, that is, spot and futures prices along with these forecasts lead to median profits that are undominated by other feature combinations, which is consistent with the hypothesis that futures prices and analyst forecasts account for factors that affect prices. We thus use this feature combination for our performance analysis.

The median profits of DDA-LDR range between 2.1% and 2.4% (of the perfect foresight value) for different feature choices, while the RH median profit is 12%. That is, despite being a data-driven policy, DDA-LDR performs even worse than RH. In contrast, we observe that DDA-SP generates median profits between 15.7% and 26.7% and also improves on the 25th percentile of profits (i.e., downside). Hence, both regularization in training and the structure encoded in these data-driven policies can help to improve on the performance of RH profits as well as the downside risk profile of the profit distribution. In addition, the DDA-SP median profits change significantly when forward-looking profits based on futures prices are considered during the training procedure, indicating that our extension of existing DDA approaches can add value. The difference between DDA-SP and RH policies are 35.9%, 19.8%, 11.2%, −4%, 19.5%, and 5%, respectively, on the copper, gold, crude oil, natural gas, corn, and soybean instances. The performance of RH and DDA-SP thus varies significantly across commodities with DDA-SP exhibiting good overall performance and RH remaining a strong contender in particular for natural gas.

Our findings advance the state-of-the-art for commodity storage operations. The extended DDA and structured policies highlight potential opportunities to enhance storage software by considering generalization error. In particular, although we apply our framework for spot trading, it can be extended to evaluate and handle generalization error when combined with forward trading.

1.1 | Related work and novelty

Our models, methods, and findings extend the literature on commodity storage, data-driven optimization, and commodity finance as discussed below.

The literature on commodity storage dates back to the warehouse management problem introduced by Cahn (1948) and further studied by Charnes and Cooper (1955), Bellman (1956), and Dreyfus (1957). The storage assets in these very early papers were managed under deterministic prices. Charnes et al. (1966) and Secomandi (2010) consider the stochastic version of the storage problem with and without rate constraints, respectively, and characterize the optimal policy. Significant recent effort has gone toward using approximate dynamic programming techniques to find near-optimal policies to the intractable storage SDP in the full-information setting (Cruise et al., 2019; Lai et al., 2010; Nadarajah et al., 2015; Nadarajah & Secomandi, 2018; Nascimento & Powell, 2008; Wu et al., 2012). Secomandi et al. (2015) consider the impact of choosing an incorrect number of factors in a prespecified price model on storage valuation and hedging. They term this price-model error. Secomandi (2015) and Nadarajah and Secomandi (2018) argue in single and network storage settings, respectively, that the RH policy is price model error-free as it uses only market futures prices as input. However, they do not analyze the impact of futures prices providing poor forecasts of the spot price as they work under the risk-neutral measure where the expected spot price equals the futures price. In summary, the extant storage literature has not empirically studied the impact of generalization error on the storage operating policy or developed data-driven operating policies that target this error. Our backtest of RH, development of DDA approaches that leverage known policy characterizations, and related empirical insights are novel to this literature. Moreover, our use of regularization and policy structure provides an ML and optimization-inspired view of managing storage operations, which is relevant beyond this setting to other real options involving commodities such as soybean, corn, and palm (Boyabatlı et al., 2017; Devalkar et al., 2011, 2018; Goel & Tanrisever, 2017) and energy (Nadarajah & Secomandi, 2021).

Our work builds on methodological work from empirical optimization (Bartlett & Mendelson, 2006; Esfahani et al., 2018) and the emerging data-driven optimization literature (see, e.g., Ban et al., 2018; Bertsimas & Kallus, 2020; Curtis & Scheinberg, 2017; Elmachtoub & Grigas, 2022), which addresses generalization error by explicitly focusing on out-of-sample performance. Strictly speaking, our paper belongs to the growing literature (e.g., Ban & Rudin, 2019; Chenreddy et al., 2019; Mandl & Minner, 2020) that empirically tests the value of data-driven optimization in operations management problems. Data-driven optimization has been applied to single-period inventory control or newsvendor applications (Ban & Rudin, 2019) and in multiperiod settings using linear or piece-wise linear decision rule approximations

(Ben-Tal et al., 2005; See & Sim, 2010), for instance, for financial contracting (Mandl & Minner, 2020). In a marketing setting, Chenreddy et al. (2019) combine ERM with polynomial approximations and inverse reinforcement learning. The forward-looking DDA that we propose extends the backward-looking DDAs in this literature. While linear decision rules are known, our assessment of their performance for commodity storage, especially when trained using estimates of future profits is new. Our structured data-driven policy and the evaluation of the value of enforcing policy structure are both novel. In addition, the parameters of the structured policies that we train are thresholds, which are easily interpretable by managers. Our models for training these data-driven policies add to the literature on interpretable ML, an area that has studied several applications ranging from classification to healthcare (see Lakkaraju & Rudin, 2017, and references therein) but none that share the structure of the commodity storage application. More broadly, our empirical finding that enforcing policy structure can improve out-of-sample performance of data-driven policies is relevant for other operations management problems where characterizations of the optimal policy structure are known.

Finally, our results contribute to recent work in commodity finance that brings to light the value of features for price prediction (Alquist & Kilian, 2010; Cortazar et al., 2018; Heath, 2019). These papers emphasize the importance of the true distribution of spot prices (as opposed to risk-neutral distributions), which is consistent with our focus. However, the aforementioned papers take a statistical view and do not focus on decision making, while we take an ML perspective and train operating policy parameters. Therefore, our comparison of DDA approaches and feature selection in the context of storage decisions add novel components to this literature. Our forward-looking DDA shows how the presence of financial markets allows one to obtain future profit estimates that can be leveraged as part of the training objective. We also assess the values of futures prices and analyst forecasts of spot prices as features when training decision rules for storage to be significant. In particular, while futures prices may provide poor forecasts of spot prices, they nevertheless provide valuable information to train policies. This finding motivates further research on the differential impact of data on prediction versus decision making.

2 | COMMODITY STORAGE OPERATIONS AND POLICY PERFORMANCE

In Section 2.1, we present a feature-based extension of the well-known storage SDP. In Section 2.2, we describe the statistical perspective used to evaluate storage policies in the literature and make a case for the value in using an ML perspective instead.

2.1 | Feature-based storage MDP

We extend the (stochastic) commodity storage problem formulated by Charnes et al. (1966), Secomandi (2010), and Lai et al. (2010). Consider a single-item, multiperiod, discrete-time, periodic-review inventory replenishment problem at a single commodity storage asset (e.g., warehouse) with a finite planning horizon T. Periods $t = 0, 1, 2, \dots, T$ equal decision stages and might correspond to hours, days, weeks, or months. The storage asset state is described by I_t and denotes the amount stored at the beginning of t. I_t is bounded by warehouse capacity C, that is, $0 \le I_t \le C$. The holding cost per unit of time and unit of inventory is denoted by $c_t^{\mathbf{h}} \ge 0$. We denote by $y_t^{\mathbf{i}} \ge 0$ the period t injection quantity and by $y_t^{\mathbf{o}} \ge 0$ withdrawal quantity at this period. These decisions are subject to injection and withdrawal limits $G^{\mathbf{i}}$ and $G^{\mathbf{o}}$, respectively. Storage operations have associated operational frictions. Specifically, injections and withdrawals incur marginal costs of $c^{\mathbf{i}} \ge 0$ and $c^{\mathbf{o}} \ge 0$, respectively, and have associated losses of $\eta^{\mathbf{i}} \in (0,1]$ and $\eta^{\mathbf{o}} \in (0,1]$. The commodity spot price in period t is denoted by p_t. The friction-adjusted purchase and selling prices are

$$p_t^{\mathbf{i}} = \frac{1}{\eta^{\mathbf{i}}} p_t + c^{\mathbf{i}}, \qquad p_t^{\mathbf{o}} = \eta^{\mathbf{o}} p_t - c^{\mathbf{o}}, \tag{1}$$

where it is common to assume that storage losses are paid in-kind, that is, using a fraction of the physically traded commodity. Note that $p_t^{\mathbf{i}} \ge p_t^{\mathbf{o}}$. As is standard in the merchant operations literature, we assume that the merchant is a price taker (i.e., injections and withdrawals do not affect the spot price). We also assume the merchant has access to the spot market only (physical trading, rather than financial trading via futures contracts).

Injection and withdrawal decisions at each period are conditioned on the information available to the user (i.e., the MDP state). Let $\xi_t := \{I_t, X_t\}$ denote all information available to the merchant at the beginning of period $0 \le t \le T$. The inventory level I_t is endogenous information as past injections and withdrawals determine its value. The remaining component is a vector of N features ($X_{t,n} \in \mathcal{X}_n, n = 1, \dots, N$), which is exogenous information and unaffected by storage operations. At period t, the distribution of the (random) spot price $p_\tau, \tau > t$ depends on X_t. Examples of features include current and past spot prices, prices of futures contracts, and investor forecasts of spot prices. Given I_t at period t, the feasible injection and withdrawal set $\mathcal{Y}_t(I_t)$ is defined as

$$\mathcal{Y}_t(I_t) := \left\{ \left(y_t^{\mathbf{o}}, y_t^{\mathbf{i}}\right) | y_t^{\mathbf{o}} \in [0, \min\{I_t, G^{\mathbf{o}}\}], y_t^{\mathbf{i}} \in [0, \min\{C - I_t, G^{\mathbf{i}}\}] \right\}. \tag{2}$$

Storage results in inventory I_t transitioning to $I_{t+1} = I_t - y_t^{\mathbf{o}} + y_t^{\mathbf{i}}$. Under nonzero marginal costs, it is easy to verify that it is suboptimal to inject and withdraw in the same period.

A storage operating policy π is a collection of decision rules $\{Y_t^\pi, t = 0, \ldots, T\}$, where $Y_t^\pi := (Y_t^{\pi,\mathbf{o}}, Y_t^{\pi,\mathbf{i}})$ is a function that assigns a pair of withdrawal and injection decisions $(y_t^{\mathbf{o}}, y_t^{\mathbf{i}})$ to each state (I_t, X_t) at period t. Denoting by Π the set of all operating policies, the value of optimally managing storage starting from a state (I_t, X_t) at period t is

$$\text{F-MDP} \qquad V_t(I_t, X_t) := \max_{\pi \in} \sum_{\tau=t}^{T} \mathbb{E} \left[\left(p_\tau^{\mathbf{o}} Y_\tau^{\pi,\mathbf{o}} - p_\tau^{\mathbf{i}} Y_\tau^{\pi,\mathbf{i}} - c_\tau^{\mathbf{h}} I_\tau^\pi \right) | X_t \right], \tag{3}$$

where $V_t(I_t, X_t)$ is the value function at period t and state (I_t, X_t), I_t^π is the inventory level reached at period t when using policy π, and $\mathbb{E}$ is expectation with respect to the true (and potentially unknown) stochastic process driving the features. We suppress the discount factor without loss of generality as it can be factored into the prices and holding cost.

An optimal policy to the storage MDP can be sequentially computed using the following stochastic dynamic programming recursion:

$$\text{F-SDP} \qquad V_t(I_t, X_t) = \max_{(y_t^{\mathbf{o}}, y_t^{\mathbf{i}}) \in \mathcal{Y}_t(I_t)} \times \left\{ p_t^{\mathbf{o}} y_t^{\mathbf{o}} - p_t^{\mathbf{i}} y_t^{\mathbf{i}} - c_t^{\mathbf{h}} I_t \; \mathbb{E}\left[V_{t+1}(I_t - y_t^{\mathbf{o}} + y_t^{\mathbf{i}}, X_{t+1}) | X_t \right] \right\}, \tag{4}$$

$\forall t = 0, \ldots, T-1$ and (I_t, X_t). Unlike the feature-based SDP presented here, the extant storage literature predefines the feature vector X_t and assumes a stochastic process for its evolution. A popular choice for X_t is the forward curve, that is, $X_t = (f_{t,t}, f_{t,t+1}, \ldots, f_{t,T})$, where $f_{t,t'}$, $t' > t$, is the time t price of a futures contract maturing at time t' and $f_{t,t} = p_t$ (see, e.g., Lai et al., 2010). The stochastic process driving these prices typically has multiple factors and satisfies $\mathbb{E}[p_{t'} | f_{t,t'}] = f_{t,t'}$. This is true in complete markets under the risk-neutral measure, where market participants have different risk preferences but attribute a unique value to the asset. However, market incompleteness is common in commodity markets and this assumption may not hold.

The structure of the optimal policy known in the commodity storage literature (see, e.g., Secomandi et al., 2015) extends to F-SDP as stated in Proposition 1 under Assumption 1.

Assumption 1 (**Bounded spot price expectation**). Assume that for all stages $t = 0, 1, \ldots, T$ it holds that

$$\mathbb{E}_t[p_{t+1} | X_t] < \infty. \tag{5}$$

Proposition 1 (**Optimal Policy Structure**). *The following holds under Assumption 1:*

(*a*) *Suppose $G^{\mathbf{i}} = G^{\mathbf{o}} = C$. Then there is an optimal policy of F-SDP and price threshold functions $P_t(X_t)$ such that at each period t and state (I_t, X_t), the optimal storage injection and withdrawal decisions satisfy*

$$\left(y_t^{\mathbf{i}}, y_t^{\mathbf{o}}\right) = \begin{cases} (C - I_t, 0) & \text{if } p_t^{\mathbf{i}} < P_t(X_t), \\ (0, 0) & \text{if } p_t^{\mathbf{o}} < P_t(X_t) \leq p_t^{\mathbf{i}}, \\ (0, I_t) & \text{if } P_t(X_t) \leq p_t^{\mathbf{o}}. \end{cases} \tag{6}$$

(*b*) *Suppose $\min\{G^{\mathbf{i}}, G^{\mathbf{o}}\} < C$. Then there is an optimal policy of F-SDP and injection and withdrawal base-stock–level functions $S_t^{\mathbf{i}}(X_t)$ and $S_t^{\mathbf{o}}(X_t)$, respectively, with $S_t^{\mathbf{i}}(X_t) \leq S_t^{\mathbf{o}}(X_t)$ such that at each period t and state (I_t, X_t) the optimal storage injection and withdrawal decisions satisfy:*

$$\left(y_t^{\mathbf{i}}, y_t^{\mathbf{o}}\right) = \begin{cases} (\min\{S_t^{\mathbf{i}}(X_t) - I_t, G^{\mathbf{i}}\}, 0) & \text{if } I_t < S_t^{\mathbf{i}}(X_t), \\ (0, 0) & \text{if } S_t^{\mathbf{i}}(X_t) \leq I_t \leq S_t^{\mathbf{o}}(X_t), \\ (0, \min\{I_t - S_t^{\mathbf{o}}(X_t), G^{\mathbf{o}}\}) & \text{if } I_t > S_t^{\mathbf{o}}(X_t). \end{cases} \tag{7}$$

We omit the proof of Proposition 1 as it follows standard reasoning available in the literature. When Assumption 1 holds, the value function can be shown to be bounded following the arguments in Lemma B.1 of Nadarajah and Secomandi (2018). The remaining parts of the proof to establish policy structure mirror Lemma B.2 of Secomandi et al. (2015).

Proposition 1(a) summarizes the bang-bang structure of the optimal policy when the storage asset is fast, that is, it has full operational flexibility (FF) and no rate constraints. In this case, the optimal policy is based on the value taken by a state-dependent price threshold $P_t(X_t)$ in relation to the friction-adjusted spot prices. Depending on the value of $P_t(X_t)$ the optimal decision is to (i) fill storage, (ii) do nothing, or (iii) empty storage. The optimal policy for a slow storage asset with rate constraints, which we refer to as limited operational flexibility (LF), is shown in Proposition 1(b). Injection and withdrawal depend on comparing inventory level with state-dependent injection and withdrawal base-stock levels. These decisions (i) fill up storage to the injection base-stock level, (ii) do nothing, or (iii) decrease inventory down to the withdrawal base-stock level.

2.2 | Policy performance evaluation

Solving F-MDP directly is challenging since we do not have a feature representation $\mathbf{X}$ or knowledge of the stochastic process M driving its evolution. We refer to $(\mathbf{X}, M)$ as the feature–model pair. Even if the feature representation was known, the computational burden of solving F-MDP is prohibitive due to the well-known curses of dimensionality. Therefore, it is common to forgo finding an optimal policy and instead solve a tractable optimization model that approximates F-MDP and delivers a heuristic policy. Given such a heuristic policy $\hat{\pi}$, its performance needs to be evaluated. We discuss the evaluation procedure used extensively in the merchant storage literature (and more broadly in stochastic optimization) and then present a data-inspired evaluation procedure, also highlighting its implications to methods that compute policies. This subsection will form the conceptual basis for our empirical results and methods in the remaining parts of the paper.

The literature on storage operations evaluates the performance of a heuristic policy $\hat{\pi}$ via simulation. Let $V^{\hat{\pi}}(p)$ be the value of applying the decisions of policy $\hat{\pi}$ on the spot-price trajectory $p := (p_t, t = 0, \dots, T)$. Further, we denote by $\pi^*_{\mathbf{X},M}$ an optimal policy to F-MDP formulated using $(\mathbf{X}, M)$. The goal is to evaluate the exact optimality gap:

$$OPT_{\mathbf{X},M}(\hat{\pi}) := \mathbb{E}_{\mathbf{X},M}\left[V^{\pi^*_{\mathbf{X},M}}(p) - V^{\hat{\pi}}(p)|X_0\right], \tag{8}$$

where $\mathbb{E}_{\mathbf{X},M}$ is expectation w.r.t. model M over feature trajectories X under feature representation $\mathbf{X}$. Since $\pi^*_{\mathbf{X},M}$ is unknown, it is common to replace it by a computable upper bound $U_{\mathbf{X},M}(X_0)$ such that $U_{\mathbf{X},M}(X_0) \geq \mathbb{E}_{\mathbf{X},M}[V^{\pi^*_{\mathbf{X},M}}(p)|X_0]$. A common upper bounding approach is information relaxation and duality (see Brown et al., 2010, for details). The resulting optimality gap estimate is

$$U_{\mathbf{X},M}(X_0) - \mathbb{E}_{\mathbf{X},M}\left[V^{\hat{\pi}}(p)|X_0\right]. \tag{9}$$

The evaluation of the optimality gap in the literature is tied to the feature representation and stochastic model assumptions. This estimate of policy performance can be misleading if the assumed pair $(\mathbf{X}, M)$ is different from the true $(\mathbf{X}^*, M^*)$. We term this potential difference between assumed and true feature–model pairs as information inconsistency. To illustrate, consider policies $\hat{\pi}^A$ and $\hat{\pi}^B$ evaluated using the exact optimality gap (8). Then it is possible that $OPT_{\mathbf{X},M}(\hat{\pi}^A) > OPT_{\mathbf{X},M}(\hat{\pi}^B)$ while $OPT_{\mathbf{X}^*,M^*}(\hat{\pi}^A) < OPT_{\mathbf{X}^*,M^*}(\hat{\pi}^B)$. Hence, the simulation-based performance ranking of these policies may differ from their ranking on real data due to information inconsistency.

Motivated by the above observation, we consider evaluating the performance of policies in a data-driven manner. Our starting point is the following definition of idealized generalization error used in reinforcement learning (see Murphy, 2005, section 4):

$$GE^*(\hat{\pi}) := OPT_{\mathbf{X}^*,M^*}(\hat{\pi}) = \mathbb{E}_{\mathbf{X}^*,M^*}\left[V^{\pi^*_{\mathbf{X}^*,M^*}}(p) - V^{\hat{\pi}}(p)|X_0\right]. \tag{10}$$

Intuitively, this definition is an assessment of how the notion of approximate optimality underlying the problem solved to obtain $\hat{\pi}$ "generalizes" to handle the exact optimality associated with F-MDP formulated with $(\mathbf{X}^*, M^*)$, which gives rise to $\pi^*_{\mathbf{X}^*,M^*}$. While conceptually appealing, similar to the issue in (8), $\pi^*_{\mathbf{X}^*,M^*}$ is unknown and hence the value $V^{\pi^*_{\mathbf{X}^*,M^*}}(p)$ too. We thus replace this value by the perfect foresight value $V^{PF}(p)$ obtained by optimizing storage operations with knowledge of the true spot prices p. Clearly this value does not depend on $(\mathbf{X}, M)$, that is, unlike $U_{\mathbf{X},M}(X_0)$ used to obtain (9) starting from (8), the term $V^{PF}(p)$ is a feature–model pair independent upper bound. The resulting computable generalization error is

$$\mathbb{E}_{\mathbf{X}^*,M^*}\left[V^{PF}(p) - V^{\hat{\pi}}(p)|X_0\right]. \tag{11}$$

We replace the expectation $\mathbb{E}_{\mathbf{X}^*,M^*}$ by its sample average approximation based on H trajectories of observed data $p^h := (p^h_0, \dots, p^h_T)$ for $h = 1, \dots, H$ to obtain the empirical generalization error

$$GE(\hat{\pi}) := \frac{1}{H}\sum_{h=1}^{H}\left[V^{PF}(p^h) - V^{\hat{\pi}}(p^h)\right]. \tag{12}$$

Minimizing $GE(\hat{\pi})$ to find a policy $\hat{\pi}^{GE} := \arg\min_{\pi\in\Pi} GE(\pi)$ is equivalent to maximizing empirical performance on observed data, that is, $\hat{\pi}^{GE}$ solves $\max_{\pi\in\Pi}\sum_{h=1}^{H} V^{\hat{\pi}}(p^h)/H$. In other words, unlike the focus of the existing storage literature on finding policies with low optimality gaps (e.g., less than a few percent) under a potentially incorrect model $(\mathbf{X}, M)$, the effort when using (12) for evaluation is redirected to ranking policies based on their performance on data. In addition to providing a data-driven ranking of policies, $GE(\hat{\pi})$ measures the empirical performance of a policy relative to the perfect foresight value. This difference is insightful, especially in volatile commodity markets, as it shows the value that can be gained from perfect knowledge of future information and has been considered in the literature (Kleindorfer et al., 2012).

Minimizing $GE(\hat{\pi})$ mitigates the possibility of incorrectly ranking policies during evaluation and concluding that a policy with poor empirical performance is near-optimal owing to the previously discussed information inconsistency between an assumed feature–model pair $(\mathbf{X}, M)$ and the true such pair $(\mathbf{X}^*, M^*)$. Although we focus mainly on performance evaluation of a given policy here, it is important to note that information inconsistency can also cause the method that computes $\hat{\pi}$ to incorrectly rank policies and thus choose one with poor performance on data. Specifically, the generalization error of a policy $\hat{\pi}$ can be larger than $\hat{\pi}^{GE}$ due to this information inconsistency. It is also common for methods to optimize over a smaller policy class $\hat{\Pi} \subset \Pi$ to achieve tractability. This restriction can result in $\hat{\Pi}$ excluding the optimal policy or more importantly near-optimal ones, and thus increase generalization error even in the absence of information inconsistency. We refer to this difference in policy sets as structural inconsistency. In summary, a firm can compare the performance of policies using traditional simulation assuming a feature–model pair and re-evaluating these policies using generalization error as the metric on data. If the ranking of policies changes, this is a signal that there is information inconsistency. When designing data-driven policies that attempt to reduce generalization error, one needs to be cognizant of both information and structural inconsistencies.

3 | RH BACKTEST

In this section, we perform an extensive backtest to evaluate the performance of RH based on generalization error as defined in (12). We describe RH in Section 3.1. We overview the data set used for our backtest in Section 3.2 and present results in Section 3.3.

3.1 | Algorithm

RH, which is sometimes referred to as forward dynamic optimization (Eydeland & Wolyniec, 2003, p. 355), is a sequential RH and a type of certainty-equivalent control that determines injection and withdrawal decisions by solving an "intrinsic" linear program formulated using point estimates of future spot prices. It does not require any training, which makes its implementation easy. We denote the time t point estimate of the spot price p_τ with $\tau > t$ by $f_{t,\tau}$ and define the vector $F_t := (f_{t,\tau}, \tau \in \mathcal{T}_t)$ where $\mathcal{T}_t = \{t, t+1, \dots, T\}$. Thus, the estimates of $p_t^{\mathbf{i}}$ and $p_t^{\mathbf{o}}$ are $f_{t,\tau}^{\mathbf{i}} = \frac{1}{\eta^{\mathbf{i}}} f_{t,\tau} + c^{\mathbf{i}}$ and $f_{t,\tau}^{\mathbf{o}} = \eta^{\mathbf{o}} f_{t,\tau} - c^{\mathbf{o}}$, respectively. We denote by $\mathcal{F}(I, t, T)$ the feasible set of inventory levels and storage decisions $\{(y_\tau^{\mathbf{i}}, y_\tau^{\mathbf{o}}, I_\tau), \tau \in \{t, t+1, \dots, T\}\}$ over a planning period $\{t, t+1, \dots, T\}$ with a starting inventory level of I. This set is defined by the following constraints:

$$I_{\tau+1} = I_\tau - y_\tau^{\mathbf{o}} + y_\tau^{\mathbf{i}}, \qquad \forall \tau \in \{t, t+1, \dots, T-1\}, \tag{13}$$

$$0 \le y_\tau^{\mathbf{i}} \le C - I_\tau, \qquad \forall \tau \in \{t, t+1, \dots, T\}, \tag{14}$$

$$0 \le y_\tau^{\mathbf{o}} \le I_\tau, \qquad \forall \tau \in \{t, t+1, \dots, T\}, \tag{15}$$

$$y_\tau^{\mathbf{i}} \le G^{\mathbf{i}}, \qquad \forall \tau \in \{t, t+1, \dots, T\}, \tag{16}$$

$$y_\tau^{\mathbf{o}} \le G^{\mathbf{o}}, \qquad \forall \tau \in \{t, t+1, \dots, T\}. \tag{17}$$

Constraints (13) model the inventory transitions. Constraints (14)–(17) enforce the restrictions on the injection and withdrawal amounts.

The ILP at period t is

$$\max_{\{(y_\tau^{\mathbf{i}}, y_\tau^{\mathbf{o}}, I_\tau), \tau \in \mathcal{T}_t\}} \sum_{\tau \in \mathcal{T}_t} \left[f_{t,\tau}^{\mathbf{o}} y_\tau^{\mathbf{o}} - f_{t,\tau}^{\mathbf{i}} y_\tau^{\mathbf{i}} - c_t^{\mathbf{h}} I_\tau \right] \tag{18}$$

$$\text{s.t. } \{(y_\tau^{\mathbf{i}}, y_\tau^{\mathbf{o}}, I_\tau), \tau \in \mathcal{T}_t\} \in \mathcal{F}(\bar{I}_t, t, T). \tag{19}$$

The objective function (18) maximizes the profit from storage operations estimated using the spot price forecast F_t with decisions subject to operational constraints $\mathcal{F}(\bar{I}_t, t, T)$.

The RH policy is based on solving ILP (18)–(19) at each stage. To elaborate, ILP is solved in the current period t given the forecast F_t and the inventory state information I_t to obtain injection and withdrawal decisions for each future period, that is, $(y_\tau^{\mathbf{i}}, y_\tau^{\mathbf{o}})$ for $\tau \in \mathcal{T}_t$. The period t decision pair $(y_t^{\mathbf{i}}, y_t^{\mathbf{o}})$ is the decision implemented by the RH policy at state (I_t, F_t). Then an ILP is formulated in period $t+1$ using updated inventory state information $I_{t+1} = I_t - y_t^{\mathbf{o}} + y_t^{\mathbf{i}}$ and an updated forecast F_{t+1}. Solving the resulting ILP gives the period $t+1$ RH decision and so on. RH thus side-steps the curse of dimensionality involved in tackling F-SDP by solving LPs based on point forecasts.

RH is popular for managing natural gas storage, where $f_{t,\tau}$ is chosen to be the time t price of a futures contract with maturity at time τ. Choosing $f_{t,\tau}$ as a futures price is directly applicable for operating storage assets of other commodities with traded futures contracts. When a futures market is absent, $f_{t,\tau}$ could be a point forecast or a prediction of p_τ.

When F-SDP is formulated for a commodity with a futures market (i.e., $X_t = F_t$), the existing literature that uses RH cites two advantages. The first is that the RH policy is consistent with the structure of an F-SDP optimal policy, which from our discussion in Section 2.2 implies that the structural component of generalization error is zero for RH. The second advantage is that RH is model-free. Specifically, the futures prices in F_t are available from traded contracts in the market and not based on any statistical model. Under the risk-neutral measure typically used in the literature, this model-free definition of $f_{t,\tau}$ as a futures price also results in an unbiased estimator of p_τ since we have $f_{t,\tau} = \mathbb{E}[p_\tau]$ for all $\tau \in \mathcal{T}_t$. However, the RH policy is applied under the real-world measure (often referred to as the physical measure) that drives spot prices, where a futures price may provide a poor forecast of the spot price. In other words, although RH is model-free, it can (and likely will) have a nonzero informational component of generalization error. To assess the generalization error and performance of RH on real data, we perform a backtest in Section 3.2.

3.2 | Data and instances

Our RH backtest is based on spot and futures price data between 2000 and 2017 for the following six commodities: copper, gold, crude oil, natural gas, corn, and soybean. Futures contracts for metals and energy are traded at the New York Mercantile Exchange (NYMEX) and for agricultural commodities at the Chicago Board of Trade (CBOT). We consider futures prices for the first 12 maturities, that is, 1- to 12-months-ahead contracts, and use monthly prices at the first trading day of the corresponding month. Even though contracts beyond 1 year are available for various commodities, these markets are typically highly illiquid with only very few contracts traded, which implies that the predictive content for

TABLE 1 Commodity spot and futures price data from Thomson Reuters (2000–2017)

Commodity	Price quotation	Annualized volatility	Spot market (Data source)	Futures market (Data source)
Metals				
Copper	USD/lb	25%	Nevada Copper (NCUCASH)	COMEX (HGc1-12)
Gold	USD/ounce	17%	Gold (XAU=)	COMEX (GCc1-12)
Energy				
Crude oil	USD/bbl	37%	West Texas Intermediate (CRUDOIL)	NYMEX (CLc1-12)
Natural gas	USD/mmbtu	65%	Henry Hub Natural Gas (NATLGAS)	NYMEX (NGc1-12)
Agricultural				
Corn	USc/bushel	29%	No.2 Yellow Corn (CORNUS2)	CBOT (Cc1-12)
Soybean	USc/bushel	25%	No.1 Yellow Soybean (SOYBEAN)	CBOT (Sc1-12)

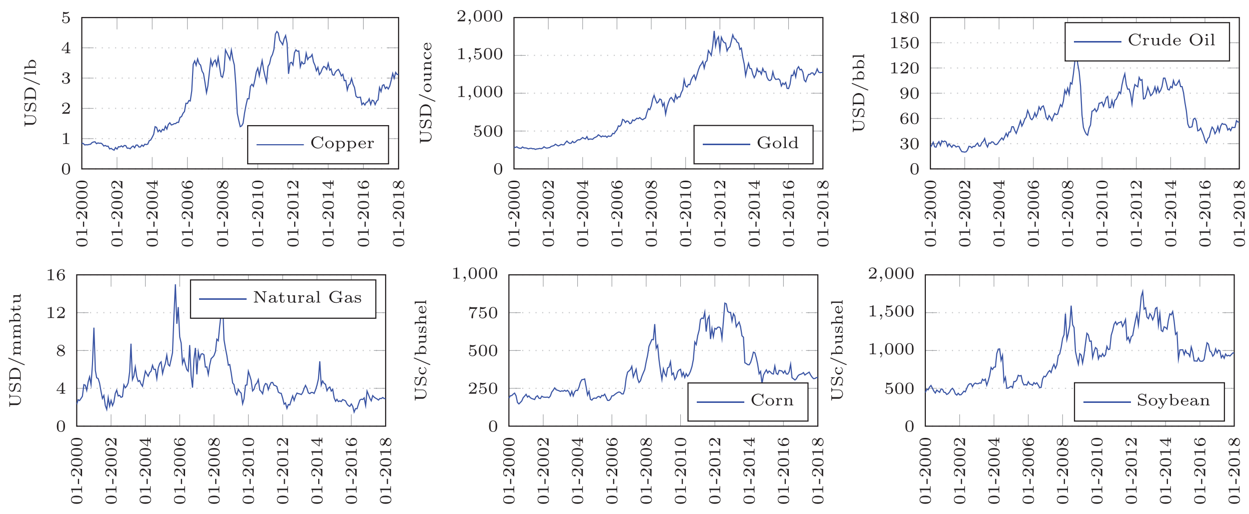

FIGURE 1 Spot prices for copper, gold, crude oil, natural gas, corn, and soybean from 2000 to 2017 [Color figure can be viewed at wileyonlinelibrary.com]

future spot prices might be low (Alquist & Kilian, 2010). Furthermore, our perfect foresight analysis on the empirical data indicates that planning horizons that are significantly smaller than 12 months are sufficient for optimal first-stage decisions (see Figure EC.1 of the Supporting Information).

Among the six commodities we consider, four of them have futures contracts with monthly maturities. The exceptions are CBOT corn and soybean futures. The former futures mature in March, May, July, September, and December while the latter futures mature in January, March, May, July, August, September, and November (www.cmegroup.com). To obtain monthly corn and soybean futures prices we employ linear interpolation following Nadarajah and Secomandi (2018) who apply the approach described in Guthrie (2009). Table 1 summarizes the sources we use to obtain data. Figure 1 plots the spot prices for each commodity.

For each commodity, we consider various operational settings for the storage asset in our backtest. Table 2 summarizes the parameters that we vary to obtain $4 \times 2 \times 3 \times 9 = 216$ instances per commodity (i.e., 1296 instances in total). Across all instances, we normalize warehouse capacity to $C = 1$, choose initial inventory $I_0 = 0$, and set storage holding cost $c_t^{\mathbf{h}} = 0$ (although, we tested plausible values for this parameter and found that it led to similar results). There are no injection and withdrawal costs ($c^{\mathbf{i}} = c^{\mathbf{o}} = 0$). Furthermore, we distinguish between fully flexible storage (FF) with $G^{\mathbf{i}} = G^{\mathbf{o}} = C$ and limited flexible storage (LF) with $G^{\mathbf{i}} = G^{\mathbf{o}} = \frac{1}{2}C$.

TABLE 2 Summary of the numerical design

Planning horizon	$T \in \{1, 3, 6, 12\}$
Storage flexibility	$G^{\mathbf{i}} = G^{\mathbf{o}} \in \{0.5, 1\}$
Operational frictions	$\eta^{\mathbf{i}} = \eta^{\mathbf{o}} \in \{1, 0.995, 0.99\}$
Subperiods	2000–2002, 2002–2004, …, 2016–2017

3.3 | Results

Our implementation of RH assumes monthly inventory review periods so that each storage decision period coincides with a futures contract maturity. This assumption is consistent

TABLE 3 Performance of futures-based RH in $V^{\mathrm{RH}}/V^{\mathrm{PF}} \cdot 100\%$ across all instances

	Mean	Min	25%-Q	50%-Q	75%-Q	Max	Mean[a]	Mean[b]
Copper	1.3	−58.5	−12.2	0.0	10.6	70.0	4.1	1.5
Gold	10.0	−107.5	0.0	0.0	23.2	78.7	9.4	11.1
Crude oil	−5.1	−171.9	−30.1	7.1	26.3	70.5	−3.3	7.6
Natural gas	18.6	−125.9	6.0	25.6	44.8	79.5	21.2	28.5
Corn	16.7	−44.3	−5.8	14.5	42.2	64.9	20.0	19.4
Soybean	24.5	−31.6	−1.0	20.8	45.6	79.8	28.1	27.3
Overall	11.0	−171.9	−4.6	9.5	34.9	79.8	13.3	15.9

[a]Mean w/o 2008–2009 (financial crisis).
[b]Mean w/o 2014–2015 (oil price drop).

with past studies of RH (see, e.g., Lai et al., 2010; Secomandi, 2015). We also note that futures markets can be more liquid than spot markets, which are often thinly traded (Geman & Smith, 2013). We tested RH based on trading in the futures market with the closest expiry (the so-called front-month contract) as a proxy for the spot price. As the results were similar, we do not report them in the paper.

3.3.1 | Empirical performance of RH

Table 3 reports statistics of the RH backtest across the 1296 instances summarized in Table 2. We obtain an assessment of the true generalization error by measuring $\frac{V^{\mathrm{RH}}}{V^{\mathrm{PF}}} \cdot 100\%$ where V^{RH} and V^{PF} are the RH and perfect foresight profits, respectively. V^{PF} is calculated based on past data via (18)–(19) given known price trajectories rather than forecasts.

We find that RH achieves 11.0% of the perfect foresight value on average. Its performance across commodities varies significantly. For instance, the mean profits vary from −5.1% for crude oil to 24.5% for soybean. The negative mean profit for crude oil was intriguing. Further investigation showed that the RH policy results in negative profits on 30.1% of the instances. Negative profits usually occur whenever the forecast shows decreasing prices and therefore the RH policy sells available inventory to the market disregarding that the purchase costs were higher. Indeed, in practice, these negative profits can be avoided by trading in forward contracts. We also observe that the financial crisis of 2008–2009 and the oil price drop during 2014–2015 significantly impact performance. Notably, if we exclude the instances corresponding to the 2014–2015 subperiod, the average performance of RH for crude oil increases from −5.1% (unprofitable storage) to 7.6% (profitable storage) and its worst-case performance improves from −171.9% to −64.2%.

The performance of RH varies with the operational storage parameters $(G^{\mathbf{i}}, G^{\mathbf{o}}, \eta^{\mathbf{i}}, \eta^{\mathbf{o}})$. We report these results in Table EC.1 of the Supporting Information. While for a given operational setting (e.g., copper, $n = 12$, $\eta^{\mathbf{i}} = \eta^{\mathbf{o}} = 0.995$) RH yields positive mean profit for a fully flexible (FF) storage asset, this average profit becomes negative once the storage asset has limited flexibility (LF), that is, once the injection and withdrawals are constrained. Moreover, if frictions are large (i.e., small $\eta^{\mathbf{i}}$ and $\eta^{\mathbf{o}}$) relative to the (expected) price changes, the warehouse slows down its activity (see, for example, the instances of gold with zero mean profit in Table S1).

The small RH profit percentages relative to the perfect foresight solution are not themselves concerning because our benchmark is anticipative but it does raise the question of whether RH can be improved. Note that this question does not arise in the RH performance results reported for natural gas in the literature (see, e.g., Lai et al., 2010), which are performed under statistical model assumptions and show that RH is within a few percent of the optimal policy value. These differences in the assessment of RH suggest that information inconsistency may be at play here but confirming this suspicion requires comparing against a method that targets generalization error, which will be the focus of Sections 4 and 5.

3.3.2 | Performance impact of the planning horizon

To understand if the performance of RH can be improved, we define variants of RH that solve an ILP at each stage formulated over a shorter horizon than $T = 12$, that is, we exclude futures prices with further maturities. In particular, we consider $T \in \{1, 3, 6, 12\}$ and use RH_{T} to denote the RH variant with a planning horizon of T periods.

Figure 2 shows that the performance of RH is sensitive to the planning horizon T on almost all instances. Exceptions include the FF instances without frictions where it is known that a one-period look-ahead policy is optimal (see related results in Table EC.1 of the Supporting Information). Using longer planning horizons in RH helps improve the average performance for crude oil, gold, and soybean but hurts profits for natural gas and corn, while this effect is mixed for copper. Several performance changes are substantial with T. For instance, the average profit percentage for natural gas reduces from over 25% for RH_1 to less than 20% for RH_{12}. A

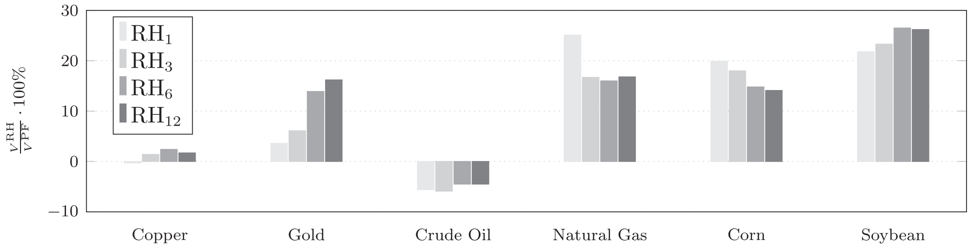

FIGURE 2 Average performance of RH$_T$ for different planning horizons T

TABLE 4 Dominance matrix of RH$_T$ for different planning horizons T reporting the percentage of instances (N= 1296) in which the RH variant in the row strictly outperforms its variant in the column

	RH$_1$	RH$_3$	RH$_6$	RH$_{12}$
RH$_1$	–	28.7	28.4	29.9
RH$_3$	31.2	–	9.6	14.8
RH$_6$	38.9	21.9	–	8.0
RH$_{12}$	41.4	27.5	12.0	–

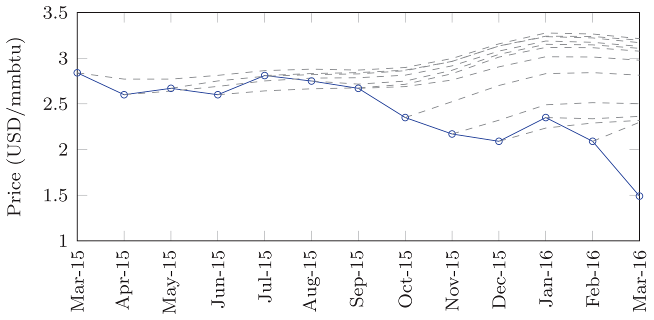

FIGURE 3 NYMEX futures curves (dashed) and realized spot prices (○) for natural gas (prices refer to closing prices at the first trading day of the corresponding month) [Color figure can be viewed at wileyonlinelibrary.com]

similar trend can be observed for corn. These results indicate that it is possible to significantly improve the performance of the standard RH policy (i.e., RH$_{12}$).

We investigate the results of Figure 2 further in the dominance matrix shown in Table 4. The one-step look-ahead policy based on RH$_1$ outperforms RH policies with longer planning horizons on a significant number of instances.

Specifically, RH$_1$ strictly improves RH$_{12}$ on 29.9% of the instances (and weakly in 58.6% of the instances, which is omitted in the paper). Further, we observe that RH$_6$ is equal to or better than RH$_{12}$ on 88.0% of the instances, which suggests that futures prices with later maturities adversely affect the performance of the RH$_{12}$ operating policy. This finding is qualitatively different from the literature on forecast horizons (Chand et al., 2002) applied to RH under a risk-neutral measure (see Section 2 for a related discussion), which states that adding futures price information to RH can only be beneficial to its performance and ignoring futures prices beyond a certain maturity does not affect performance (Cruise et al., 2019). One possible explanation for the longer futures maturities hurting the performance of RH in our real-world backtest could be related to these futures prices providing a poor forecast of the corresponding spot price as exemplarily shown in Figure 3.

3.3.3 | Value of reoptimization

The preceding qualitative deviation from the literature also brings into question whether there is value in the reoptimization of ILP, which is needed to define the RH policy. Under the risk-neutral measure and standard statistical model assumptions, reoptimization has been shown to add significant value over the intrinsic (static) policy based on the forward curve available at the initial stage (Lai et al., 2010; Secomandi, 2015). We assess if this remains the case in our backtest. We define the value of reoptimization as VReO := $(\frac{V^{\mathrm{RH}}-V^{\mathrm{ILP}}}{V^{\mathrm{PF}}}) \cdot 100\%$ with V^{ILP} denoting the profit obtained using the ILP. While RH revises the injection and withdrawal decision plan in each period based on new futures price information and updated inventory, ILP determines the plan for all periods based on F_0. Figure 4 summarizes the value of reoptimization for RH with different planning horizons. The value of reoptimization can be either positive or negative. Across the instances, this value is negative for RH$_{12}$ on 37.0% of the instances, which is significant. Figure EC.2 of the Supporting Information shows in more detail when reoptimization would have generated positive value and when not. We observe that ILP outperforms RH especially in phases of sharp price jumps or drops, once again indicating that the inability of futures prices to forecast spot price changes can lead to the behavior of RH for spot trading when evaluated on real data being substantially different from what has been observed in controlled simulations.

3.3.4 | Value of perfect price information

In Section EC.2.3 of the Supporting Information, we investigate what information (albeit idealistic) could be provided to RH in lieu of futures prices to improve its performance. Our results show that one-step-ahead spot price information generates significant additional profits compared to standard RH with futures price information. The perfect foresight value for

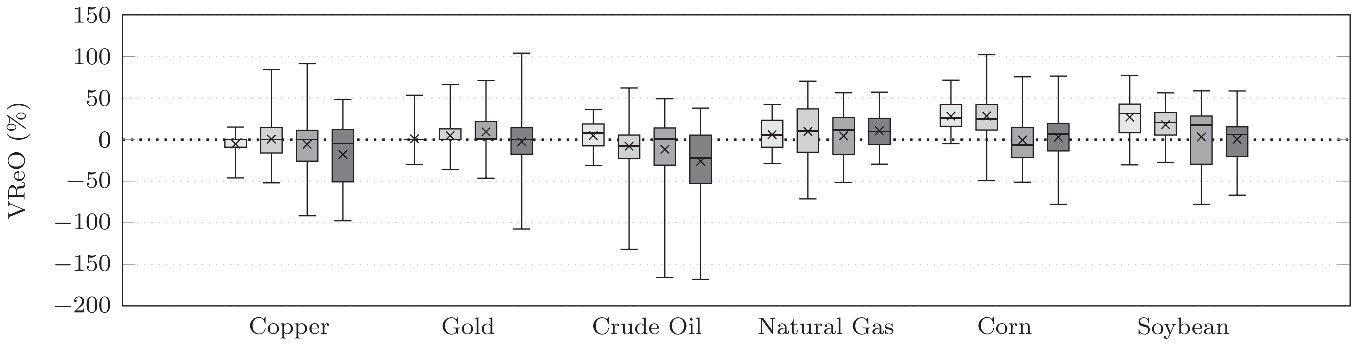

FIGURE 4 Value of reoptimization for $T \in \{1, 3, 6, 12\}$ (light gray to dark gray). Note on boxplot characteristics: minimum, 1st-, 2nd-, 3rd-quartile, maximum

flexible storage assets can almost fully (on average 98.5%) be captured by correctly classifying the direction of one-step-ahead price movements. The improvement for limited flexibility is over 50%.

Therefore, we show that there is opportunity to improve on RH for spot trading. This observation motivates the development of DDAs for managing storage that targets generalization error.

4 | DATA-DRIVEN DECISION RULES

In this section, we focus on data-driven decision rules for managing commodity storage. In Section 4.1, we present a forward-looking DDA. We apply this approach using linear decision rules and structured policies in Section 4.2 and Section 4.3, respectively. Finally, we discuss robustness aspects of these policies in Section 4.4.

4.1 | Forward-looking policy training and evaluation framework

We consider the computation of data-driven injection and withdrawal decision rules $(u_t^{\mathbf{i}}(X, \beta^{\mathbf{i}}), u_t^{\mathbf{o}}(X, \beta^{\mathbf{o}}))$ at each stage, which are functions of features X and parameter vectors $\beta^{\mathbf{i}}$ and $\beta^{\mathbf{o}}$. The decision rule parameters $\beta^{\mathbf{i}}$ and $\beta^{\mathbf{o}}$ are the trained coefficients of features that fully characterize the decision rules for purchasing and selling (see Equations (28) and (30)). They are computed using the ERM framework (Vapnik, 1998, p. 32). Suppose we have a historical spot price sample path covering T' periods. This procedure divides this sample path into training, validation, and testing segments corresponding to the subperiods $\{0, \dots, T^{\mathrm{s}}\}$, $\{T^{\mathrm{s}}+1, \dots, T^{\mathrm{v}}\}$, and $\{T^{\mathrm{v}}+1, \dots, T'\}$, respectively. The coefficients of the decision rules are chosen to maximize regularized profit on the training sample path segment. Regularization is added to avoid overfitting decisions to a single sample path (see Mohri et al., 2012, p. 28). The standard backward-looking math program—see Bartlett and Mendelson (2006), and for recent applications, Ban and Rudin (2019) and Mandl and Minner (2020)—used for training is

$$\max_{\substack{\{(y_t^{\mathbf{i}}, y_t^{\mathbf{o}}, I_t), t\in\{0,1,\dots,T^{\mathrm{s}}\}\}\\ \{(\beta_n^{\mathbf{o}}, \beta_n^{\mathbf{i}}), n=0,\dots,N\}}} \frac{1}{T^{\mathrm{s}}}\left[\sum_{t=0}^{T^{\mathrm{s}}}(p_t^{\mathbf{o}} y_t^{\mathbf{o}} - p_t^{\mathbf{i}} y_t^{\mathbf{i}} - c^h I_t)\right] - \lambda \|(\beta^{\mathbf{o}}, \beta^{\mathbf{i}})\|_1 \tag{20}$$

$$\text{s.t.}\ \ \{(y_t^{\mathbf{i}}, y_t^{\mathbf{o}}, I_t), t \in \{0, 1, \dots, T^{\mathrm{s}}\}\} \in \mathcal{F}(\bar{I}_0, 0, T^{\mathrm{s}}), \tag{21}$$

$$y_t^{\mathbf{i}} = u_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}}), \forall t \in \{0, 1, \dots, T^{\mathrm{s}}\}, \tag{22}$$

$$y_t^{\mathbf{o}} = u_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}}), \forall t \in \{0, 1, \dots, T^{\mathrm{s}}\}, \tag{23}$$

where the first term of the objective is the average profit over the training period and the second term is a 1-norm regularization where $\lambda \geq 0$ controls the weight of this term in the objective. Constraint (21) enforces operational constraints while constraints (22)–(23) encode the decision rule structure. Given β coefficients, a data-driven storage policy π^{DDA} is the collection of feasible injection and withdrawal decision rules $\{(y_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}}), y_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}})), t \in \mathcal{T}_0\}$, where $y_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}}) := \max\{\min\{C - I_t, G^{\mathbf{i}}, u_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}})\}, 0\}$ and $y_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}}) = \max\{\min\{I_t, G^{\mathbf{o}}, u_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}})\}, 0\}$. Math program (20)–(23) is solved several times by varying $\lambda \geq 0$. Each solution produces a possibly different set of β coefficients and corresponding policy. Among these policies, π^{DDA} is chosen to be the one that results in the largest profit on the validation segment. Finally, the profit of π^{DDA} is evaluated on the testing segment against the perfect foresight solution.

An important property of (20)–(23) is that all the data used in its definition are available at or before period T^{s}. Hence, it is common in the literature to maximize the profit over the training period. In the context of commodity storage as well as other applications with financial markets, forward-looking market information beyond the training set is available via futures prices at period T^{s}. As already discussed in earlier sections, the futures price $f_{t,\tau}$ may be a reasonable predictor for the spot price p^{τ} at least for near-term maturities. Based on this observation, we consider solving the following

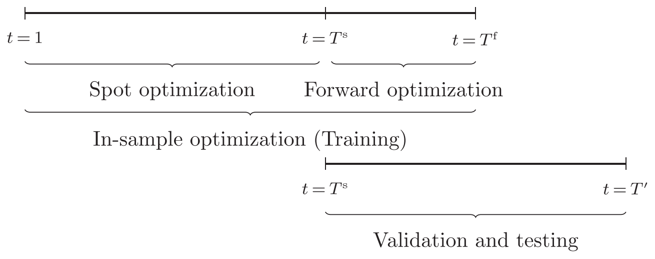

FIGURE 5 Optimization (training) and evaluation framework

forward-looking training math program over $T^f > T^s$ periods:

$$\max_{\substack{\{(y_t^{\mathbf{i}},y_t^{\mathbf{o}},I_t),t\in\{0,1,\dots,T^f\}\}\\ \{(\beta_n^{\mathbf{o}},\beta_n^{\mathbf{i}}),n=0,\dots,N\}}} \frac{1}{T^f}\left[\underbrace{\sum_{t=1}^{T^s}(p_t^{\mathbf{o}}y_t^{\mathbf{o}} - p_t^{\mathbf{i}}y_t^{\mathbf{i}} - c_h I_t)}_{\text{Standard ERM}} + \underbrace{\sum_{t=T^s+1}^{T^f}(f_{T^s,t}^{\mathbf{o}}y_t^{\mathbf{o}} - f_{T^s,t}^{\mathbf{i}}y_t^{\mathbf{i}} - c_t^{\mathbf{h}}I_t)}_{\text{Forward optimization: Estimate of future profit}}\right] - \lambda\|(\beta^{\mathbf{o}},\beta^{\mathbf{i}})\|_1 \tag{24}$$

$$\text{s.t. } \{(y_t^{\mathbf{i}},y_t^{\mathbf{o}},I_t),t\in\{0,1,\dots,T^f\}\}\in\mathcal{F}(\bar{I}_0,0,T^f), \tag{25}$$

$$y_t^{\mathbf{i}} = u_t^{\mathbf{i}}(X_t,\beta^{\mathbf{i}}), \forall t\in\{0,1,\dots,T^f\}, \tag{26}$$

$$y_t^{\mathbf{o}} = u_t^{\mathbf{o}}(X_t,\beta^{\mathbf{o}}), \forall t\in\{0,1,\dots,T^f\}. \tag{27}$$

Constraints (25)–(27) are analogous to (21)–(23) but defined over a longer time horizon T^f. The objective function (24) has an extra term not found in (20). This term corresponds to the estimated profits over the periods $\{T^s+1,\dots,T^f\}$ obtained using futures prices available at T^s. Indeed, choosing $T^f = T^s$ gives us back the standard ERM math program. We illustrate our forward-looking framework to train storage policies in Figure 5.

The data-driven framework described above can be used to target generalization error, which as discussed in Section 2.2 can be viewed as having components due to information inconsistency and structural inconsistency. The effect of the former inconsistency can be mitigated via feature selection in the context of the training, validation, and testing framework. The effect of latter inconsistency depends on the choice of $(u_t^{\mathbf{i}}(X,\beta^{\mathbf{i}}),u_t^{\mathbf{o}}(X,\beta^{\mathbf{o}}))$, which will be the focus of Sections 4.2 and 4.3. In this paper, we evaluate if our forward-looking ERM adds value compared to using the standard ERM math program.

4.2 | Linear decision rules (DDA-LDR)

The common choice for decision rules is an affine mapping of features to decisions, referred to as linear decision rules (LDRs; see Ban & Rudin, 2019, for a newsvendor example and for additional references). Such a mapping for $(u_t^{\mathbf{i}}(X_t,\beta^{\mathbf{i}}),u_t^{\mathbf{o}}(X_t,\beta^{\mathbf{o}}))$ at period t is:

$$u_t^{\mathbf{i}}(X_t,\beta^{\mathbf{i}}) := \sum_{n=0}^{N}\beta_n^{\mathbf{i}}X_{t,n}, \quad u_t^{\mathbf{o}}(X_t,\beta^{\mathbf{o}})) := \sum_{n=0}^{N}\beta_n^{\mathbf{o}}X_{t,n}, \tag{28}$$

where $\beta_n^{\mathbf{i}}\in\mathcal{B}\subset\mathbb{R}$ and $\beta_n^{\mathbf{o}}\in\mathcal{B}\subset\mathbb{R}$ are feature coefficients that are unknown to the merchant and must be learned from historical time series data. To allow for a feature-independent intercept, we set $X_{t,0}=1\ \ \forall t\in\mathcal{T}_0$. The linear parameterization of $u_t^{\mathbf{o}}$ and $u_t^{\mathbf{i}}$ is not very restrictive since one can introduce new features that are nonlinear functions of the original features. For instance, such functions could involve interactions terms (e.g., $\beta_3^{\mathbf{i}}X_{1t}X_{2t}$), polynomials (e.g., $\beta_1^{\mathbf{i}}X_{1t}^2$), and lagged observations (e.g., $\beta_2^{\mathbf{i}}X_{1,t-1}$).

We refer to the policy obtained based on the choice (28) as DDA-LDR. A computational advantage of DDA-LDR is that (24)–(27) becomes a linear program that is efficient to solve. In terms of structural consistency, an LDR will in general not have the same structure as an optimal storage policy. Thus, it may suffer from generalization error because of this inconsistency, which motivates the structured decision rules considered next.

4.3 | Structured decision rules (DDA-SP)

We choose $u_t^{\mathbf{i}}(X_t,\beta^{\mathbf{i}})$ and $u_t^{\mathbf{o}}(X_t,\beta^{\mathbf{o}})$ guided by the policy structures outlined in Proposition 1.

We begin by considering the optimal policy structure in the full flexibility case (i.e., Proposition 1(a)), which is based on a price threshold $P_t(X_t)$. Our goal will be to compute $P_t(X_t)$ using feature information in a data-driven manner and impose the optimal policy structure on the decision rules that we compute. In other words, we enforce

$$P_t(X_t) := \sum_{n=1}^{N}\beta_n X_{t,n}, \tag{29}$$

and choose

$$(u_t^{\mathbf{i}}(X_t,\beta),u_t^{\mathbf{o}}(X_t,\beta)) = \begin{cases} (C-I_t,0) & \text{if } p_t^{\mathbf{i}} < \sum_{n=1}^{N}\beta_n X_{t,n},\\ (0,0) & \text{if } p_t^{\mathbf{o}} < \sum_{n=0}^{N}\beta_n X_{t,n} \le p_t^{\mathbf{i}},\\ (0,I_t) & \text{if } \sum_{n=0}^{N}\beta_n X_{t,n} \le p_t^{\mathbf{o}}. \end{cases} \tag{30}$$

The choice for $u_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}})$ and $u_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}})$ in (30) is fundamentally different from a linear decision rule, as these decisions are not directly parameterized by features but features specify thresholds (as decision signals) within the optimal policy structure. Math program (24)–(27) under definition (30) has an mixed-integer program representation, which is detailed in Section EC.3.1 of the Supporting Information. This representation facilitates the use of off-the-shelf commercial solvers for solving this math program.

For a storage asset with limited flexibility, the estimation of a single price threshold is not sufficient because at the same market price p_t, different purchase-and-inject and withdraw-and-sell decisions $y_t^{\mathbf{i}}$ and $y_t^{\mathbf{o}}$, respectively, can be optimal depending on the current inventory level I_t. We specify $u_t^{\mathbf{i}}(X_t, \beta^{\mathbf{i}})$ and $u_t^{\mathbf{o}}(X_t, \beta^{\mathbf{o}})$ using the policy structure of Proposition 1(b) and the parameterized base-stock levels

$$S_t^{\mathbf{i}}(X_t) := \max\left\{0, \sum_{n=0}^{N} \beta_i^{\mathbf{i}} X_{t,n}\right\},$$

$$S_t^{\Delta}(X_t) := \max\left\{0, \sum_{n=0}^{N} \beta_i^{\Delta} X_{t,n}\right\}, \tag{31}$$

with $S_t^{\mathbf{o}}(X_t) := S_t^{\mathbf{i}}(X_t) + S_t^{\Delta}(X_t)$. This additive formulation of the base-stock levels is required to ensure that $S_t^{\mathbf{i}} \leq S_t^{\mathbf{o}}$. We provide a mixed-integer linear program to compute the $\beta_i^{\mathbf{i}}$ and β_i^{Δ} coefficients in Section EC.3.2 of the Supportingx Information.

4.4 | Robustness of DDA-SP

Storage policies need to account for price and estimation risk. Price risk arises because commodity prices are uncertain, while estimation risk is a consequence of errors incurred when determining the parameters of a policy. DDA-SP accounts for both these risks as discussed below.

Differences between in-sample and out-of-sample profits of a storage policy can be attributed to the informational and structural components of generalization error (discussed in Section 2.2). The informational component arises due to commodity spot prices in the test set being uncertain and different from the training set (i.e., price risk). Regularization adds bias to the estimator to improve out-of-sample performance by avoiding overfitting (Mohri et al., 2012).

It can also be viewed as ensuring that the policy is trained using a robust objective in the forward-looking math program (20)–(23). To understand this, note that setting λ equals zero in the objective (20) amounts to maximizing profits on the training set without accounting for price risk. For a positive λ, the bias added by the regularization makes this objective robust, that is, training and validation procedures used to determine policy parameters account for spot prices differing from historical prices within some uncertainty set (see, e.g., Gao et al., 2017, for theoretical results on the robustness interpretation of regularization).

The structural component of generalization error, interestingly, has implications on both model complexity and the impact of estimation error. Consider DDA-LDR, which is inconsistent with the optimal policy structure in general. The complexity of the class of policies represented by LDRs is a function of the richness of features. For instance, if features used in the definition (28) of LDR include the class of prespecified threshold and/or base-stock policies, then the class of LDRs subsume the set of structured policies considered by DDA-SP. In contrast, regardless of the richness of features, the class of policies that DDA-SP considers is restricted to those satisfying optimal policy structure, thus potentially reducing model complexity relative to LDRs. Additionally, policy structure makes DDA-SP robust to estimation error. To see this, consider DDA-LDR again. Changes in feature values X_t directly translate into changes in the injection or withdrawal decisions for LDRs as seen in (28). In contrast, for the choice (30) of DDA-SP, small changes in feature values may not affect decisions if the estimated $P_t(X_t)$ value remains in the same interval as the exact threshold. Analogous reasoning holds for the double-base stock structure in the case of a slow storage asset.

We numerically verify robustness of DDA-SP compared to DDA-LDR in Section 5.3.

5 | PERFORMANCE EVALUATION OF DDA

In this section, we evaluate the performance of DDA-LDR/SP compared to ILP and RH.

5.1 | Setup

Table 5 summarizes the approaches we compare and the data that they exploit. In addition to the data considered in the RH backtest (Section 3.2), we also include analyst forecast data based on a feature selection study detailed in Section 5.4, which shows that spot prices, futures prices, and median analyst forecast constitute an undominated feature combination. We use *Bloomberg's Analysts' Median Composite Forecast* that reports the median of the price forecasts offered by up to 31 major financial institutions. While individual expert forecasts may exhibit high prediction errors, by using the median forecast over a variety of well-established financial institutions, we expect some error diversification (Cortazar et al., 2018). Based on the median forecasts, we generate monthly analyst forecast curves $A_t = (a_{t,\tau} : \tau \in \mathcal{A} = \{t, t+1, \ldots, t+T\})$ for the six commodities for planning horizons up to 12 months for the limited time period of 2008 to 2017 (Note: The RH results based on analyst forecasts and AR(1) spot history rather than futures curves are reported in Tables EC.2 and EC.3 of the Supporting Information). Our experimental setup for operational parameters is identical to the first three rows of Table 2. To obtain instances spanning multiple subperiods, we split the data as shown in Table 6. This yields $2 \times 3 \times 8 \times 4 = 192$ instances per commodity.

TABLE 5 Benchmarks and feature data

Approach	Computation	Policy consistency	Futures prices	Spot history	Analyst forecasts
ILP	LP	Yes	✓	No	No
RH	LP	Yes	✓	No	No
DDA-LDR	LP	No	✓	✓	✓[a]
DDA-SP	MILP	Yes	✓	✓	✓[a]

aData availability allows analyst forecasts to be used from Jan 2008.

TABLE 6 Backtesting setup

Sample	In-sample	Out-of-sample	Sample	In-sample	Out-of-sample
1	2000–2001	2002–2003	5	2008–2009	2010–2011
2	2002–2003	2004–2005	6	2010–2011	2012–2013
3	2004–2005	2006–2007	7	2012–2013	2014–2015
4	2006–2007	2008–2009	8	2014–2015	2016–2017

Referring to Table 6, we optimize based on a single sample path (e.g., 2000–2001) and evaluate on a test set (e.g., 2002–2003). We repeat this procedure for all test sets on a broad variation of operational storage parameters and then show the results as mean and quartile statistics across the 192 instances. The rationale behind this is that it represents the setting for decision making in practice: The storage manager trains the policy parameters on a training set (including validation on a validation set) and evaluates on a test set.

We tested sensitivity on training horizons by evaluating DDA-SP for three different training set lengths, that is, 12 months, 24 months, and 36 months. A training length of 24 months resulted in the best performance on average. Using a shorter training cycle of 12 months or a longer training of 36 months can deteriorate the downside performance of data-driven policies and foster downside outliers by not fully capturing the underlying price behavior (too short training sets) or by training on structural breaks (too long training sets). Apart from that, median performance when employing 12 and 24 months of training is similar. For more details, we refer to Section EC.7.1 of the Supporting Information. The following results are based on 24 months of training.

We further test DDA-LDR and DDA-SP with and without forward optimization. For the forward optimization, we use F_t containing futures prices with the 12 closest monthly maturities ($T^{\mathrm{f}} = T^{\mathrm{s}} + 12$) that outperformed both a shorter forward optimization horizon of $T^{\mathrm{f}} = T^{\mathrm{s}} + 6$ and having no forward optimization, that is, $T^{\mathrm{f}} = T^{\mathrm{s}}$ (see Section EC.7.2 of the Supporting Information for more details). We report sensitivity toward frictions and storage flexibility in Supporting Information EC.7.3. The effect of discount rates on performance is reported in Supporting Information EC.7.4. To avoid overfitting and to enable feature selection, we apply regularization to regularize DDA-LDR and DDA-SP in a cross-validation procedure, which leads to better performance compared to unregularized DDA for the majority of instances (see Section EC.8 of the Supporting Information for more details).

5.2 | Performance evaluation

Figure 6 summarizes the performance results of the different storage policies. More detailed numbers are reported in the Supporting Information (TableEC. EC.4 and EC.6). Note that the mean performance in general increases by excluding subperiods 2008–2009 (financial crisis) and 2014–2015 (oil price drop), which is shown in Figure EC.5 of the Supporting Information.

(i) *DDA versus RH*.On the majority of commodities, we observe that DDA-SP can outperform RH consistently (see Table EC.6) and by a significant amount (see Table EC.4): Over all commodities, DDA-SP with forward optimization strictly dominates RH in 63.7% of the instances (Table EC.6) with a median performance of 26.7% of the perfect foresight profit, while RH achieves a median performance of 12.0% (Table EC.4). For copper, crude oil, and corn, the DDA-SP profit improvements over RH are statistically significant at the 1% level. For natural gas, DDA-SP without and with forward optimization improves the performance of RH on 66.1% and 46.4% of the instances, respectively. However, RH has a better median performance than DDA-SP. This may be reasonable in markets that are particularly efficient, which is the case for natural gas compared to less efficient metal and agricultural markets (Kristoufek & Vosvrda, 2013). In highly efficient markets, all available information is already included in the futures prices. Another reasonable explanation is the

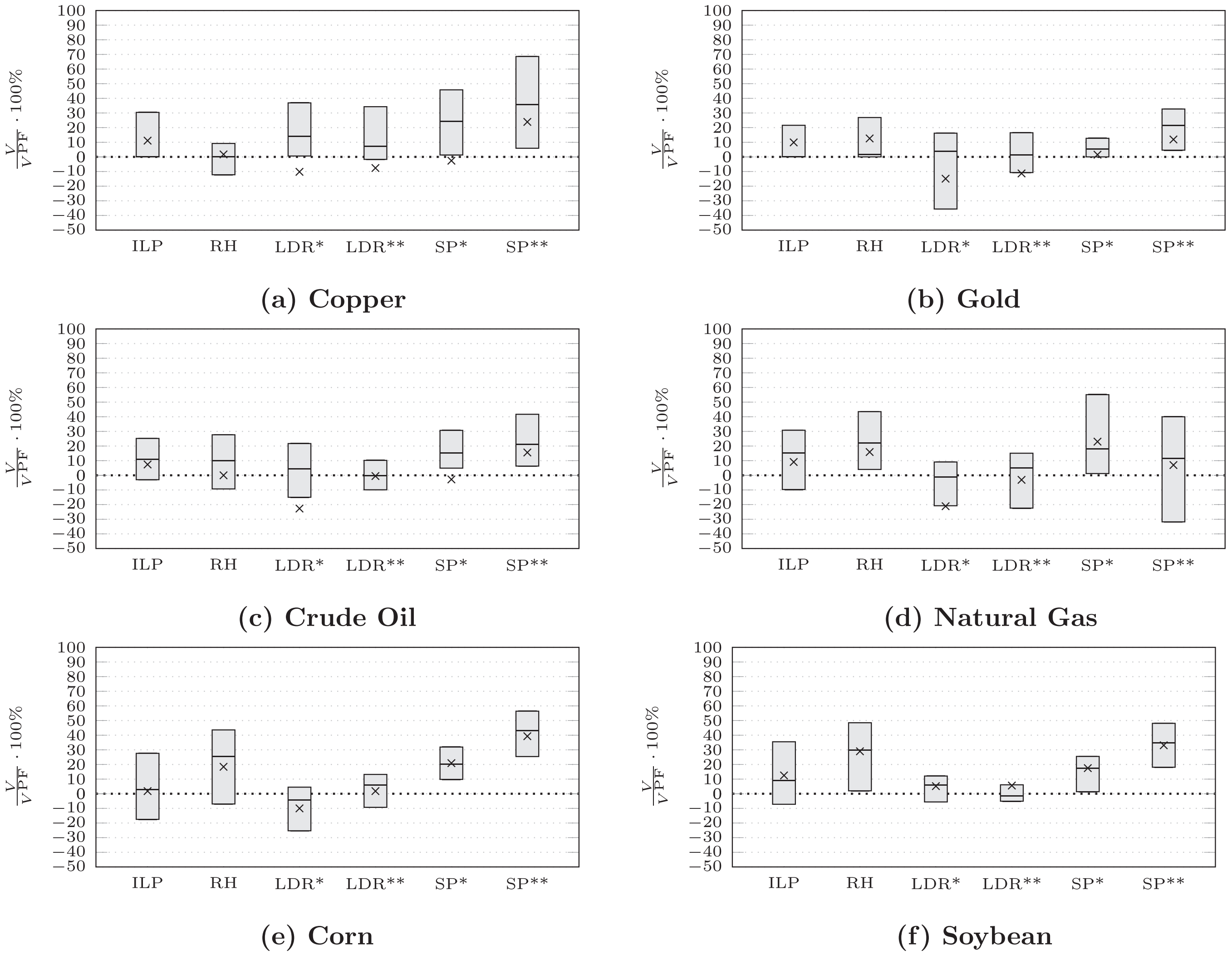

FIGURE 6 Out-of-sample performance of the different policies from 2002 until 2017 across all instances. *Note*. * w/o forward optimization, ** w/ forward optimization. Boxplots characteristics: first-, second-, third-quartile, mean (×). For a better graphical comparability of the quartiles, we do not explicitly show the whiskers in these plots. The corresponding minimum and maximum values can be found in Table EC.4 of the Supporting Information

substantially higher volatility of gas prices (65%) compared to the other commodities (17%–37%) we considered in our experiments (see Table 1). This might favor periodic reoptimization (RH) that exclusively uses forward-looking information, while feature-based DDA is sensitive to structural breaks in the training history. The observed improvement on RH using DDA-SP is in contrast to prior results in the literature that establish the near-optimality of RH. This finding confirms that the effects of information inconsistency in the evaluation of policies can be significant when assuming a feature–model pair and favors generalization error, a data-driven metric, for such evaluation. In addition to the evaluation of policies, generalization error appears to be a useful metric to focus on when computing policies, as we do for computing the DDA-SP policy.

(ii) *DDA with forward optimization versus DDA w/o forward optimization*.We observe the value of considering forward optimization in the DDA policies to be positive. For instance, including forward-looking information in DDA-SP improves average performance on 63.9% of the instances (Table EC.6). Except for natural gas, the profit improvement of DDA-SP through forward optimization is statistically significant at the 1% level. Thus, our forward-looking training approach adds value over backward-looking DDAs.

(iii) *DDA-SP versus DDA-LDR*.We observe that the LDR approach, which does not ensure policy consistency, performs poorly for our constrained multi-stage optimization problem. DDA-SP, which respects policy structure, strictly outperforms DDA-LDR on 77.6% (with forward optimization) and 73.6% (without forward optimization) of the instances, respectively (Table EC.6). The profit gain from DDA-LDR** to DDA-SP** is statistically significant at the 1% level for all commodities under consideration. These results suggest that there is significant structural inconsistency in DDA-LDR, which is reduced by imposing policy structure in DDA-SP.

TABLE 7 Performance of DDA-SP in $V/V^{\text{PF}} \cdot 100\%$ with respect to storage flexibility

	Mean	Min	25%-Q	50%-Q	75%-Q	Max
	FF					
Copper	24.1	−115.0	12.1	35.9	58.4	78.6
Gold	17.4	−65.1	8.6	21.4	32.3	65.3
Crude oil	13.3	−129.5	7.4	21.3	39.9	66.6
Natural gas	9.0	−69.3	−21.4	17.7	40.0	59.5
Corn	35.0	−6.6	27.8	39.3	49.6	64.8
Soybean	28.8	0.9	18.1	30.2	38.2	52.9
Overall	21.3	−129.5	8.9	26.6	42.9	78.6
	LF					
Copper	23.7	−188.2	5.9	33.7	70.2	83.6
Gold	6.3	−136.6	2.3	17.0	32.8	80.5
Crude oil	18.0	−159.1	4.7	22.7	45.3	75.5
Natural gas	5.0	−68.0	−34.5	1.9	39.7	71.5
Corn	43.6	−3.7	24.8	55.2	60.3	71.9
Soybean	37.5	0.0	19.2	45.5	53.6	71.1
Overall	22.4	−188.2	7.1	12.0	36.9	83.6

(iv) *DDA-SP under full and limited flexibilities.*Table 7 shows the disaggregated performance of DDA-SP** from Figure 6 with respect to storage flexibility. The results of DDA-SP relative to the perfect foresight bound are not fundamentally different between fully flexible storage assets (FF) and limited flexible storage assets (LF), that is, DDA-SP performs well for both of the storage settings. However, we observe for the FF case that it is more effective to train a price threshold P_t, rather than the more general double base-stock structure. While DDA-SP with the FF structure yields an average (median) performance of 21.3% (26.6%), DDA-SP with the LF structure yields 11.9% (22.8%). Furthermore, DDA-SP-FF is more efficient and reduces computation times of DDA-SP-LF (above 3600 s) on average by almost 90%.

5.3 | Improvement of downside risk

We investigate the performance of methods in terms of the 25%-quartile of the profit distribution on each instance, which is representative of downside risk.

Figure 7 displays these results. Despite the DDA-LDR policies being trained using regularization, their 25-th percentile of profits are worse than RH on roughly 50% of the commodities and instances. In contrast, DDA-SP policies improve on the downside risk of RH policies or are comparable for all commodities except natural gas. For natural gas, where RH was shown to be a strong competitor, the downside risk measured as the 25%-quartile performance can be improved by monthly reoptimization of DDA-SP, which increases the 25%-quartile performance from −31.8% to −14.0% of the perfect foresight value. Thus, consistent with the discussion in Section 4.4, both regularization and policy structure in DDA-SP are valuable to manage downside risk.

5.4 | Feature selection

For effectively using DDA-SP, and in particular reducing information inconsistency, selecting the right initial feature set is crucial. We consider the following candidate features: spot prices, futures prices, analyst forecasts, temperature, the S&P 500 index, and the Trade Weighted U.S. Dollar Index. We will employ as a reference the feature combination used to obtain the results in earlier sections, specifically spot prices, futures prices, and analyst forecasts. Our results show that all three feature categories were relevant for storage decisions. In addition to the feature type, the lag of features also matters (see Tables EC.16– EC.19 of the Supporting Information).

Our results reported in Supporting Information EC.7.5 show that ignoring futures and analyst forecast features from the reference feature combination and relying on a pure backward-looking approach with spot price features only deteriorates performance.

However, there can be situations when liquid futures contracts or analyst forecasts are absent. In this case, it may be worth considering other features. Table EC.12 of the Supporting Information therefore compares the performance of DDA-SP with spot price features only to DDA-SP with spot price and macroeconomic features (i.e., the S&P 500 index and the Trade Weighted U.S. Dollar Index) that have been shown to drive commodity prices. The results show that in the absence of futures and analyst forecasts, adding macroeconomic features can help in particular with respect to downside risk. This is an important result with practical implications as there are commodities where futures and analyst forecasts are not available, for example, for commodities without liquid futures markets (e.g., asphalt or specific types of polyethylene such as HDPE, LDPE, and LLDPE).

Our additional results reported in Table EC.13 of the Supporting Information also show that whenever both futures and analyst forecasts are consistently available, additional macroeconomic features do not lead to a consistent performance improvement. One reason may be that macroeconomic information is already priced into futures and analyst forecast rates (Rational Expectation Hypothesis).

The absence of analyst forecasts however deteriorates storage performance (see Table EC.14 of the Supporting Information). This observation adds to the empirical findings from Cortazar et al. (2018) by showing that analyst forecasts also improve storage decisions. Cortazar et al. (2018) find similar support for price forecasting. Specifically, adding analyst forecasts as features on top of futures prices improves spot price forecast accuracy, arguing that this improvement is likely because futures-based forecasts alone may not incorporate explicit information about the risk premium.

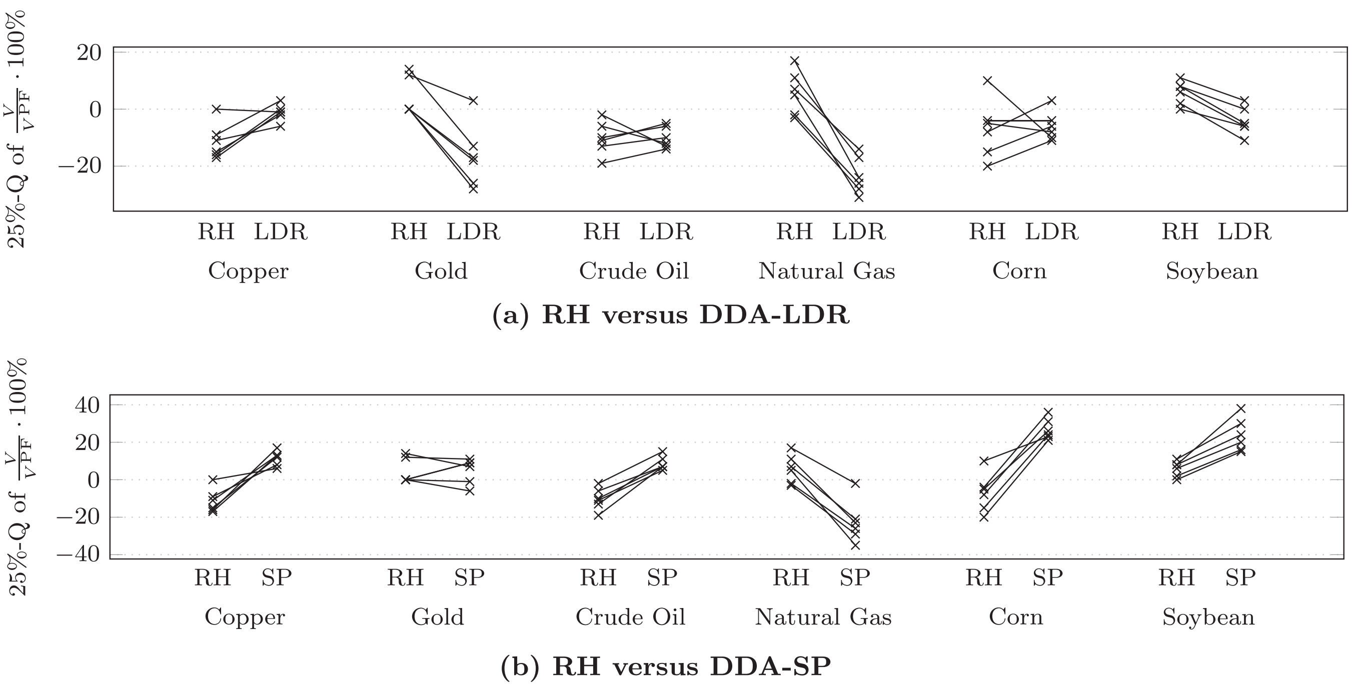

FIGURE 7 25%-quartile of $V/V^{\mathrm{PF}} \cdot 100\%$ of RH versus DDA-LDR with forward optimization and RH versus DDA-SP with forward optimization for the six combinations of $G^{\mathbf{i}} = G^{\mathbf{o}} \in \{0.5, 1\}$ and $\eta^{\mathbf{i}} = \eta^{\mathbf{o}} \in \{1, 0.995, 0.99\}$

For natural gas where DDA performs comparatively poor in our experiments, we test the effect of the additional feature *temperature* that has been shown to drive natural gas prices (see, e.g., Nick & Thoenes, 2014). Therefore, we collect monthly average temperature data and add it as an additional feature to the original DDA models. Our results from Table EC.15 of the Supporting Information confirm that temperature does not provide significant additional information for storage decisions. The results only slightly improve performance on single instances.

5.5 | Summary of insights

Our findings have implications on both storage practice and data-driven optimization research.

The existing literature evaluates the performance of the RH policy relative to the optimal policy of a storage MDP with full-information assumptions. In this setting, RH has been shown to yield near-optimal profits. However, we show that this evaluation may be misleading if applied to operate storage on real data due to generalization error. We make four related observations from Section 3.2: (i) RH can yield unprofitable storage operations ($V^{\mathrm{RH}} < 0$), (ii) ignoring futures price information can be beneficial, (iii) the value of reoptimization is not necessarily positive, and (iv) the direction (upward or downward) of the one-step-ahead price forecast is essential.

We show that there are two potential sources of generalization error: informational inconsistencies and structural inconsistencies. To mitigate the adverse effects of generalization error, we propose data-driven and ML-based policies that explore feature data (e.g., available analyst forecasts and futures prices or macroeconomic and weather features). We find that these policies can outperform RH without requiring the reoptimization of a linear program or tuning the planning horizon. Further, the linear decision rule approach from the data-driven optimization literature is not effective in our setting. Structured policies that encode properties of an optimal policy are instead needed to improve on RH. Finally, extending the standard ERM approach to include forward-looking information (if available, as is the case in commodity markets) can improve the performance of data-driven policies.

6 | CONCLUSIONS

We study the fundamental commodity storage problem. RHs are widely used in academia and practice to compute storage operating policies due to their computational attractiveness and known near-optimality in simulation experiments based on specific model assumptions. We demonstrate on real data that the empirical performance of RH can be suboptimal due to generalization error and propose a forward-looking ERM approach to compute linear decision rules and structured data-driven policies, also highlighting how it addresses informational and structural inconsistencies. We find that data-driven policies that encode an optimal policy structure exhibit robust performance across commodities and time periods in our data set, while linear decision rules perform worse than RH, despite being trained using data. In addition to uncovering the importance of policy structure in a data-driven optimization setting, using forward-looking futures price information in the training phase on top of historical spot prices can be crucial to improve out-of-sample performance. On the other hand, the additional value of learning from historical spot price data in our best data-driven storage policy compared to using only futures prices for this purpose sheds light on its performance relative to RH, which uses futures price alone. For markets such as natural gas, which are highly efficient and exhibit high volatility, the value of learning from historical spot prices appears to be limited and both RH and our DDA show good performance. In contrast, this value is substantial in less efficient and/or less volatile commodity markets such as copper, gold, crude oil, corn, and soybean, where the DDA can outperform RH.

Our DDA and structured policies advance the state-of-the-art for commodity storage. They suggest potential value in having existing software, which already incorporates back-testing capabilities, to also directly target generalization error when computing storage decisions. This research can be enhanced in several ways, of which we briefly state three. The first is to improve our backtest by allowing more granular intramonthly trading and leveraging data on trading volume to select only "liquid" futures contracts for use in RH and DDA. The second is to extend our backtest to understand the impact of generalization error and the performance of RH and DDA when forward trades are combined with spot trades. The third is to investigate DDAs that directly minimize downside risk when computing operating policies as opposed to relying on regularization for potential risk mitigation as we do in this paper.

ACKNOWLEDGMENTS

We thank the department editor Sridhar Seshadri, the senior editor, and two anonymous referees for their constructive comments to improve the paper.

ORCID

Christian Mandl https://orcid.org/0000-0001-6782-2218
Selvaprabu Nadarajah https://orcid.org/0000-0001-6218-5862
Stefan Minner https://orcid.org/0000-0001-6127-8223
Srinagesh Gavirneni https://orcid.org/0000-0002-4107-8539

REFERENCES

Alquist, R., & Kilian, L. (2010). What do we learn from the price of crude oil futures? *Journal of Applied Econometrics*, *25*(1), 539–573.

Ban, G.-Y., El Karoui, N., & Lim, A. (2018). Machine learning and portfolio optimization. *Management Science*, *64*(3), 1136–1154.

Ban, G.-Y., & Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, *67*(1), 90–108.

Bartlett, P., & Mendelson, S. (2006). Empirical minimization. *Probability Theory and Related Fields*, *3*(1), 311–334.

Bellman, R. (1956). On the theory of dynamic programming—A warehousing problem. *Management Science*, *2*(3), 272–275.

Ben-Tal, A., Golany, B., Nemirovski, A., & Vial, J.-P. (2005). Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing and Service Operations Management*, *7*(3), 248–271.

Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, *66*(3), 1025–1044.

Boyabatlı, O., Nguyen, J., & Wang, T. (2017). Capacity management in agricultural commodity processing and application in the palm industry. *Manufacturing and Service Operations Management*, *19*(4), 551–567.

Breslin, J., Clewlow, L., Elbert, T., Kwok, C., & Strickland, C. (2008). Gas storage: Overview and static valuation. *Energy Risk, November* 2008, 62–68.

Breslin, J., Clewlow, L., Elbert, T., Kwok, C., & Strickland, C. (2009). Gas storage: Rolling intrinsic valuation. *Energy Risk, January* 2009, 61–65.

Brown, D., Smith, J., & Sun, P. (2010). Information relaxations and duality in stochastic dynamic programs. *Operations Research*, *58*(4), 785–801.

Cahn, A. (1948). The warehouse problem. *Bulletin of the American Mathematical Society*, *54*(11), 1073.

Chand, S., Hsu, V., & Sethi, S. (2002). Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing and Service Operations Management*, *4*(1), 25–43.

Charnes, A., & Cooper, W. (1955). Generalization of the warehousing problem. *Operations Research Quarterly*, *6*(4), 131–172.

Charnes, A., Drèze, M., & Miller, M. (1966). Decision and horizon rules for stochastic planning problems: A linear example. *Econometrica*, *34*(2), 307–330.

Chenreddy, A., Pakiman, P., Nadarajah, S. & Chandrasekaran, R. R., A. (2019). Smoile: A shopper marketing optimization and inverse learning engine. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 2923–2942). ACM.

Cortazar, G., Millard, C., Ortega, H., & Schwartz, E. (2018). Commodity price forecasts, futures prices and pricing models. *Management Science*, *65*(9), 3949–4450.

Cruise, J., Flatley, L., Gibbens, R., & Zachary, S. (2019). Control of energy storage with market impact: Lagrangian approach and horizons. *Operations Research*, *67*(1), 1–9.

Curtis, F., & Scheinberg, K. (2017). Optimization methods for supervised machine learning: From linear models to deep learning. *INFORMS Tutorials in Operations Research*, 89–113.

Devalkar, S., Anupindi, R., & Sinha, A. (2011). Integrated optimization of procurement, processing, and trade of commodities. *Operations Research*, *59*(6), 1369–1381.

Devalkar, S., Anupindi, R., & Sinha, A. (2018). Dynamic risk management of commodity operations: Model and analysis. *Manufacturing and Service Operations Management*, *20*(2), 317–332.

Dreyfus, S. (1957). An analytic solution of the warehouse problem. *Management Science*, *4*(1), 99–104.

Elmachtoub, A., & Grigas, P. (2022). Smart "predict, then optimize." *Management Science*, *68*(1), 9–26.

Energy Quants (2018). *Energy quants – software*. http://www.energyquants.com/software/

Esfahani, P., Shafieezadeh-Abadeh, S., Hanasusanto, G., & Kuhn, D. (2018). Data-driven inverse optimization with imperfect information. *Mathematical Programming*, *167*(1), 191–234.

Eydeland, A., & Wolyniec, K. (2003). *Energy and power risk management - New developments in modeling, pricing, and hedging* (1st ed.). Wiley Finance.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (1st ed.). Springer Series in Statistics.

Gao, R., Chen, X., & Kleywegt, A. (2017). *Wasserstein distributional robustness and regularization in statistical learning*. https://arxiv.org/abs/1712.06050

Geman, H., & Smith, W. (2013). Theory of storage, inventory and volatility in the LME base metals. *Resources Policy*, *38*(1), 18–28.

Goel, A., & Tanrisever, F. (2017). Financial hedging and optimal procurement policies under correlated price and demand. *Production and Operations Management*, *26*(10), 1924–1945.

Gray, J., & Khandelwal, P. (2004a). Towards a realistic gas storage model. *Commodities Now, June* 2004, 1–4.

Gray, J., & Khandelwal, P. (2004b). Realistic gas storage models II: Trading strategies. *Commodities Now, September* 2004, 1–5.

Guthrie, G. (2009). *Real options in theory and practice*. Oxford University Press.

Heath, D. (2019). Macroeconomic factors in oil futures markets. *Management Science*, *65*(9), 3949–4450.

Kleindorfer, P., Neboian, A., Roset, A., & Spinler, S. (2012). Fleet renewal with electric vehicles at La Poste. *Interfaces*, *42*(5), 465–477.

Kristoufek, L., & Vosvrda, M. (2013). Commodity futures and market efficiency. *Energy Economics*, *42*(1), 50–57.

Kyos (2018). *Kystore —Gas storage optimization software*. https://www.kyos.com/energy-asset-optimization/gas-storage-optimization/

Lacima (2018). *Lacima analytics—Valuation and optimization suite*. https://www.lacimagroup.com/page18759/ValuationampOptimisationSuite.aspx

Lai, G., Margot, F., & Secomandi, N. (2010). An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation. *Operations Research*, *58*(3), 564–582.

Lakkaraju, H., & Rudin, C. (2017). Learning cost-effective and interpretable treatment regimes. *Proceedings of the* 20*th International Conference on Artificial Intelligence and Statistics*, *54*, 166–175.

LME (2017). *A detailed guide to the London Metal Exchange*. https://www.lme.com/-/media/Files/Brochures/Detailed-Guide-to-the-LME.pdf?la=en-GB

Mandl, C., & Minner, S. (2020). Data-driven optimization for commodity procurement under price uncertainty. *Manufacturing and Service Operations Management*, in press.

MathWorks (2018). *Natural gas storage valuation*. http://www.mathworks.com/matlabcentral/fileexchange/47667-natural-gas-storage-valuation

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press.

Murphy, S. (2005). A generalization error for q-learning. *Journal of Machine Learning Research*, *6*(37), 1073–1097.

Nadarajah, S., Margot, F., & Secomandi, N. (2015). Relaxations of approximate linear programs for the real option management of commodity storage. *Management Science*, *61*(12), 3054–3076.

Nadarajah, S., & Secomandi, N. (2018). Merchant energy trading in a network. *Operations Research*, *66*(12), 1304–1320.

Nadarajah, S., & Secomandi, N. (2021). *Real options in energy: A guided analysis of the operations literature*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3877512

Nascimento, J., & Powell, W. (2008). *Optimal approximate dynamic programming algorithms for a general class of storage problems*. Princeton University Press.

Nick, S., & Thoenes, S. (2014). What drives natural gas prices? A structural VAR approach. *Energy Economics*, *45*(1), 517–527.

Reuters (2018). *U.S. copper stocks sprint to record high, reflecting transport costs, weak demand*. https://www.reuters.com/article/us-comex-copper-stocks/u-s-copper-stocks-sprint-to-record-high-reflecting-transport-costs-weak-demand-idUSKCN1G61JU

Secomandi, N. (2010). Optimal commodity trading with a capacitated storage asset. *Management Science*, *56*(3), 449–467.

Secomandi, N. (2015). Merchant commodity storage practice revisited. *Operations Research*, *63*(5), 1131–1143.

Secomandi, N., Lai, G., Margot, F., Scheller-Wolf, A., & Seppi, D. (2015). Merchant commodity storage and term structure model error. *Manufacturing and Service Operations Management*, *17*(3), 302–320.

See, C.-T., & Sim, M. (2010). Robust approximation to multiperiod inventory management. *Operations Research*, *58*(3), 583–594.

Vapnik, V. (1998). *Statistical learning theory*. Wiley.

Williams, J., & Wright, B. (1991). *Storage and commodity markets*. Cambridge University Press.

Wu, O., Wang, D., & Qin, Z. (2012). Seasonal energy storage operations with limited flexibility: The price-adjusted rolling intrinsic policy. *Manufacturing and Service Operations Management*, *14*(3), 455–471.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Mandl, C., Nadarajah, S., Minner, S., & Gavirneni, S. (2022). Data-driven storage operations: Cross-commodity backtest and structured policies. *Production and Operations Management*, *31,* 2438–2456. https://doi.org/10.1111/poms.13683