EpiPivot: Learning to Control the Simplex Method under Epistemic Uncertainty

Guantao Zhao[1] **Mahdi Noorizadegan**[2] **Shihao Yang**[1] **Nicoleta Serban**[1]
[1]Georgia Institute of Technology [2]Northumbria University
gzhao46@gatech.edu, {shihao.yang,nserban}@isye.gatech.edu
m.noorizadegan@northumbria.ac.uk

Abstract

The simplex method is one of the most widely used algorithms for linear programming (LP), but its practical efficiency is highly sensitive to pivot selection. We propose EpiPivot, which casts pivot-rule selection as a controllable decision process under epistemic uncertainty arising from partial observations of the simplex state. EpiPivot addresses two complementary aspects of this uncertainty: temporal information captured through trajectory conditioning and joint uncertainty across candidate pivot rules represented by an Epistemic Neural Network (ENN) with shared latent sampling. At each iteration, Thompson sampling selects a pivot rule based on jointly sampled finite-horizon cost predictions. Experiments on synthetic, structured, and real-world linear programming instances demonstrate that EpiPivot reduces weighted pivot cost relative to classical and learning-based baselines, with particularly pronounced gains on structurally challenging instances.

1 Introduction

The simplex method, introduced by Dantzig in 1947 [Dantzig, 1963] and recognized among the ten most important algorithms of the twentieth century [Cipra, 2000], serves as both a foundational building block for a broad class of operations research algorithms and a core computational engine underlying modern industrial optimization. It can be viewed as a discrete, iterative search procedure over the vertices of the feasible region, where each vertex corresponds to a basic feasible solution (BFS). The algorithm proceeds through pivot operations that update the current basis while preserving feasibility and seeking objective improvement.

An ideal iterative search algorithm would progressively and controllably reduce uncertainty about the optimum. Binary search exemplifies this: each iteration deterministically eliminates half the remaining uncertainty [Nowak, 2008]. In contrast, the simplex method may follow exponentially long pivot paths under classical pivot rules in the worst case [Klee and Minty, 1972]. This difficulty stems not from individual pivot rules themselves, but from the structural complexity of the search space: different rules may induce fundamentally different trajectories, yet the long-term consequences of a pivot decision cannot be captured from the current state alone. This gap creates uncertainty about the long-term solution cost under each pivot rule: rules with similar immediate gains may lead to substantially different numbers of subsequent pivots, leaving their relative performance difficult to predict. We view this uncertainty as *epistemic*, reflecting incomplete knowledge of the mapping from the current state and rule choice to future solution cost. Even when the underlying LP and pivot operations are deterministic, the solver may remain uncertain about which rule will yield a shorter search trajectory, as illustrated in Figure 1.

In this work, we propose *EpiPivot*, which establishes a new perspective on the simplex method and addresses the challenge of pivot selection by explicitly modeling epistemic uncertainty. We identify this uncertainty as two-dimensional: a *vertical* dimension of temporal dependence across

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

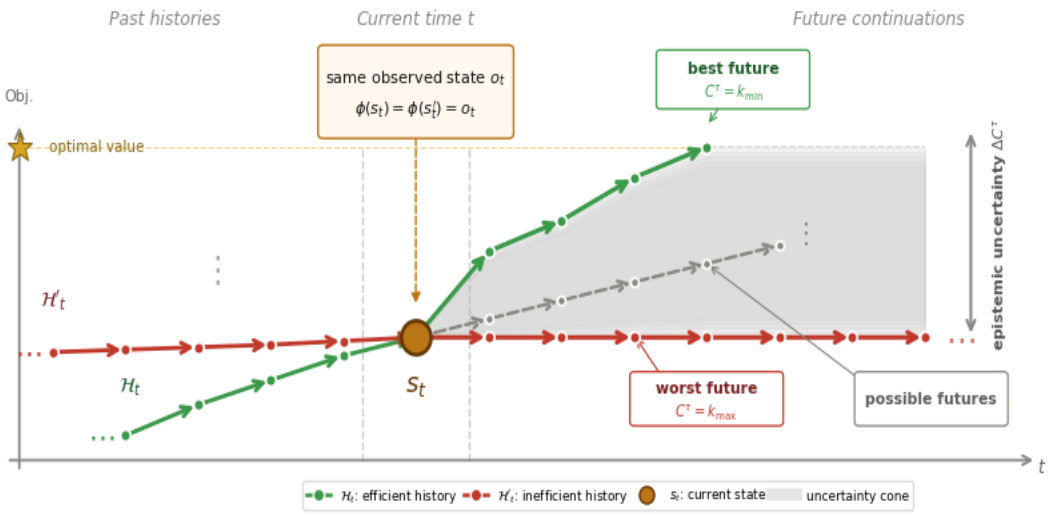


Figure 1: Illustration of the limitations of local observations in simplex pivot selection. Two distinct histories $\mathcal{H}_t$ and $\mathcal{H}'_t$ reach states s_t and s'_t with identical observed features, $\phi(s_t) = \phi(s'_t) = o_t$, while their subsequent pivot trajectories exhibit different convergence behavior. The shared observation alone does not distinguish these future outcomes. We use this example to illustrate epistemic uncertainty in assessing future search costs from the available observations. ΔC^τ denotes the difference in search costs over horizon τ. Historical trajectory information may provide additional context for these assessments.

the optimization trajectory, and a *horizontal* dimension of joint uncertainty across candidate pivot rules. To address both simultaneously, we employ a transformer-based trajectory encoder to capture path-dependent signals from recent optimization history, going beyond the Markovian assumption that the current state alone suffices for decision-making [Ma et al., 2023], and an Epistemic Neural Network [Osband et al., 2023] with a shared latent index to model the joint distribution of long-horizon costs across pivot rules, combined with Thompson sampling for robust pivot selection at decision time. This uncertainty-aware mechanism reduces simplex iterations compared to fixed pivot rules, and extensive experiments demonstrate that *EpiPivot* substantially outperforms both classical and existing learning-based methods across diverse LP instances and problem scales.

In summary, our main contributions are: (1) A new perspective on simplex search via epistemic uncertainty, identifying two dimensions: temporal path dependence and joint uncertainty across pivot rules. (2) An uncertainty-aware framework addressing both dimensions through temporal attention and ENN-based joint modeling with Thompson sampling. (3) Extensive experiments demonstrating consistent improvements over classical and learning-based baselines, with EpiPivot successfully solving instances where classical rules fail to converge. (4) New empirical insights into LP instance structure of broad relevance to the OR community.

2 Related Work

Learning-Augmented Optimization

Recent work has explored the use of ML to improve classical optimization algorithms that suffer from performance bottlenecks in complex search and decision-making. This line of research is called learning augmented optimization [Kotary et al., 2021, Bengio et al., 2021]. The key idea is not to replace the original algorithm with a learning model. Instead, machine learning is embedded within a master algorithm to support critical decisions during execution.

Within this framework, learning augmented methods have been applied to a few classical optimization algorithms. Examples include branching decisions in branch-and-bound for mixed-integer programming [Gasse et al., 2019, Khalil et al., 2016] and the selection of cutting planes in integer programming [Chételat and Lodi, 2023, Tang et al., 2020]. In these approaches, learning is used to guide internal algorithmic decisions while preserving the exactness and theoretical guarantees of the underlying optimization methods. As a result, they provide a practical way to embed learning components into classical optimization methods.

Learning-Based Pivot Selection in the Simplex Method

Recent work has explored how machine learning can be incorporated into the simplex method for linear programming. These efforts focus on pivot selection during simplex iterations, aiming to guide the search through the space of basic feasible solutions toward more efficient trajectories.

Traditional simplex implementations rely on hand-crafted pivot rules such as the Dantzig rule [Dantzig, 1963], Bland's rule [Bland, 1977], and the Steepest Edge rule [Forrest and Goldfarb, 1992]. Each of these rules is designed around a specific decision criterion and can be effective in certain situations. However, because a rule follows a fixed preference throughout the entire solution process, it cannot adapt to changes in problem structure or different stages of the algorithm. In challenging instances, relying on a single rule often leads to inefficient pivot sequences.

More recent work has attempted to learn pivoting strategies directly using machine learning. One such approach adopts imitation learning, where pivot decisions are learned from experts constructed using knowledge of the optimal basis [Liu et al., 2024]. While this approach can approximate expert behavior, the bipartite graph representation scales with problem dimensions, limiting generalization to LP instances of different sizes without retraining.

Another line of work formulates the simplex method as a Markov decision process and applies reinforcement learning (RL) to choose among existing pivot rules [Suriyanarayana et al., 2022]. This demonstrates that learning-based strategies can outperform a fixed rule in some settings. However, the approach is evaluated primarily on small-scale instances. It typically assumes a fixed problem size, as changes in the number of variables or constraints alter the underlying state and action spaces. Moreover, while value-based RL theoretically optimizes long-term cumulative return, the practical learning signal is constrained by the local state representation, limiting its capacity to capture trajectory level algorithmic behavior.

Overall, despite differences in technical approaches, existing methods largely focus on improving the quality of individual pivot decisions based on local information. How to characterize and exploit the long-term impact of pivot decisions during the simplex process, while preserving correctness and achieving good generalization, remains largely unexplored.

3 Methodology

3.1 Problem Formulation

We formulate simplex pivot-rule selection as a controllable decision process (see Appendix A.1 for the LP formulation and simplex background). Let $\xi = (A, b, c, B_0)$ denote an LP instance together with its initial Phase-II feasible basis. At iteration t, the complete simplex state is $s_t = (\xi, B_t)$, where B_t is the current basis. The controller selects a pivot rule $a_t \in \mathcal{A}$, where $\mathcal{A}$ is a set of candidate pivot rules. With fixed tie-breaking and numerical conventions, the state transition is deterministic:

$$s_{t+1} = \mathcal{T}(s_t, a_t).$$

The simplex engine retains access to the full tableau, whereas the learned controller receives only a compact observation $o_t = \phi(s_t)$ and the recent observation history

$$\mathcal{H}_t = \big(o_{\max(0,t-H+1)}, \ldots, o_t\big).$$

Thus, although the complete state determines the next state given an action, the current compact observation may not contain sufficient information to determine future pivot costs.

Given a complete state s_t and a candidate rule a, we define the finite-horizon cost

$$C^{\tau}(s_t, a) = \sum_{k=0}^{\tau-1} w(a)\, \mathbb{I}\Big[s_{t+k}^{(a)} \text{ is not optimal}\Big],$$

where $w(a)$ is the per-pivot cost, $s_t^{(a)} = s_t$ and $s_{t+k+1}^{(a)} = \mathcal{T}(s_{t+k}^{(a)}, a)$, with optimal states treated as absorbing. This cost measures the weighted pivot effort over horizon τ when rule a is applied throughout the rollout. At decision time, only one pivot is executed before the controller selects a rule again. Our goal is to reduce the overall computational cost through adaptive pivot-rule selection.

Definition 1 (Epistemic Uncertainty in the Simplex Method)**.** *Let $\mathbb{P}$ denote the probability measure induced by the LP-instance distribution and the trajectory-generation policy. For a candidate rule a,*

we define epistemic uncertainty as the conditional distribution of its finite-horizon cost given the observed history:

$$\mu_{t,a}^{\mathrm{epi}}(D \mid h) := \mathbb{P}(C^{\tau}(S_t, a) \in D \mid \mathcal{H}_t = h)\,,$$

for any measurable set $D \subseteq \mathbb{R}$. Here, S_t denotes the random complete state and h a realization of the observation history.

For a fixed complete state and rollout convention, the cost is deterministic:

$$\mathrm{Var}(C^{\tau}(S_t, a) \mid S_t) = 0.$$

However, it may remain uncertain given the controller's partial observations. This uncertainty arises from incomplete information about the complete simplex state, rather than environmental stochasticity. Since the history includes the current observation O_t, conditioning on history cannot increase the expected conditional variance:

$$\mathbb{E}[\mathrm{Var}(C^{\tau}(S_t, a) \mid \mathcal{H}_t)] \leq \mathbb{E}[\mathrm{Var}(C^{\tau}(S_t, a) \mid O_t)]\,.$$

3.2 Size-Invariant State Representation

Directly using the full tableau as input produces an input dimension that scales with problem size. We instead construct a fixed-dimensional observation $o_t = \phi(s_t)$ from statistical summaries of the simplex tableau. This representation allows the same controller to operate on LP instances of varying sizes without changing its input architecture.

Construction of State Feature The state vector is composed of statistical features derived from multiple components of the simplex tableau: reduced cost statistics characterizing the availability and strength of improving directions; right-hand side features reflecting feasibility and degeneracy; row and column norm summaries capturing geometric properties of the feasible region; and rule sensitive features encoding interactions between reduced costs and column geometry. Aggregating these yields a compact 25D size-invariant feature vector, with precise definitions provided in Appendix B.

3.3 Trajectory-Conditioned Algorithmic Cost Prediction

Under partial observability, the future computational behavior of the simplex method cannot be fully characterized by the current state alone, particularly in the presence of degeneracy and path dependence. We therefore condition cost prediction on a brief history of past states to predict the finite-horizon algorithmic cost of each candidate pivot rule, yielding a trajectory-aware control strategy.

3.3.1 Trajectory-Aware Temporal Encoding

We use the recent observation history $\mathcal{H}_t$ defined in Section 3.1 as the input sequence. To encode $\mathcal{H}_t$, we apply a multi-head self-attention mechanism over the sequence, projecting each state vector into a shared latent space and aggregating temporal patterns across attention heads. The resulting pooled embedding provides a compact, order-sensitive summary of recent optimization dynamics.

3.3.2 Finite-Horizon Cost Modeling with Epistemic Uncertainty

Given the trajectory embedding at iteration t and a candidate pivot rule a, our objective is to predict the finite-horizon cost $C^{\tau}(s_t, a)$ over horizon τ, chosen to capture meaningful differences among pivot rules beyond short-term improvement signals. Since state transitions remain deterministic, any uncertainty in predicting $C^{\tau}(s_t, a)$ is epistemic in nature, arising from partial observability rather than environmental stochasticity. We therefore characterize a distribution over cost predictions rather than relying on a single point estimate.

3.3.3 Epistemic Neural Network for Uncertainty Representation

To represent epistemic uncertainty over future pivot costs, we adopt an Epistemic Neural Network [Osband et al., 2023], which supports efficient joint predictions and is better suited for sequential decision-making than Monte Carlo dropout or deep ensembles. The ENN augments a deterministic network with a latent index variable $z \sim p(z)$, defining

$$\hat{C}^{\tau} = f_{\theta}\big(\psi_{\theta}(\mathcal{H}_t, a), z\big),$$

where $\psi_\theta(\mathcal{H}_t, a)$ denotes the action-conditioned trajectory embedding, and θ comprises the trainable parameters of the input projection, positional embeddings, temporal attention encoder, and cost-prediction head. Variability across z induces a predictive distribution over $C^\tau(s_t, a)$ reflecting epistemic uncertainty from partial observability. Different values of z correspond to alternative latent algorithmic regimes that are indistinguishable under the observed trajectory but lead to substantially different future costs. In this way, the shared latent index z simultaneously addresses both the *vertical* and *horizontal* dimensions of epistemic uncertainty without introducing artificial noise into the optimization process. The full architecture and hyperparameter details are provided in Appendix C.

At each iteration, we sample a cost realization from the predictive distribution and select the pivot rule by minimizing the sampled cost,

$$a_t = \arg\min_a \ \hat{C}_a^\tau.$$

This follows the classical Thompson sampling principle [Thompson, 1933]: rather than minimizing expected cost, each decision is conditioned on a plausible realization of future algorithmic behavior, naturally accounting for uncertainty and producing pivot choices robust to degeneracy and path dependence. In particular, since a single z is shared across all candidate rules at each iteration, the sampled cost realizations are jointly consistent, ensuring that rule comparisons are made under the same hypothesized future rather than independent marginal draws, a property that is essential for reliable pivot selection as established in Proposition 2.

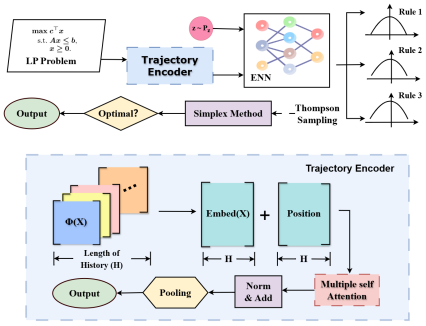


Figure 2: Overview of the *EpiPivot* framework.

The overall workflow is illustrated in Figure 2, with complete procedures detailed in Algorithms 1 and 2 in the Appendix.

3.4 Learning Objective and Supervision

Training minimizes a Huber loss between predicted and observed finite-horizon costs, averaged over multiple latent index samples per training example to robustly handle heavy-tailed cost distributions. Labels are generated via controlled simplex rollouts: for each trajectory, we simulate forward for horizon τ under each candidate rule, using the resulting pivot cost as the supervision signal and assigning a constant penalty C_{fail} to failed rollouts, with both τ and C_{fail} calibrated based on problem scale and structure. Full details are provided in Appendix C.

4 Analysis of Decision-Making under Epistemic Uncertainty

4.1 Epistemic Uncertainty in the Simplex Method

As formalized in Definition 1, deterministic simplex dynamics can still give rise to uncertainty over future pivot costs when conditioned on partial observations. The following construction illustrates how identical current summaries can correspond to different finite-horizon costs.

Proposition 1 (Pivot-Cost Ambiguity)**.** *A non-injective tableau summary need not uniquely determine the finite-horizon pivot cost, even under a fixed deterministic pivot rule. Specifically, there exist such a summary $\tilde{\phi}$, two complete simplex states $s_t \neq s_t'$, a deterministic pivot rule $a \in \mathcal{A}$, and a horizon τ such that*

$$\tilde{\phi}(s_t) = \tilde{\phi}(s_t'), \qquad C^\tau(s_t, a) \neq C^\tau(s_t', a).$$

Proof sketch. As a counterexample, let the simplified observation map $\tilde{\phi}$ record only the set of nonbasic variables with negative reduced costs. Consider two LP instances, Fast and Slow, both initialized with their slack bases. Their initial states satisfy $\tilde{\phi}(s_t^{\mathrm{F}}) = \tilde{\phi}(s_t^{\mathrm{S}})$. Under the Largest Coefficient (LC) rule with smallest-index tie-breaking, the Fast instance reaches optimality in one pivot, whereas the Slow instance first takes a degenerate pivot and reaches optimality in the second pivot. Hence, for any $\tau \geq 2$, $C^\tau(s_t^{\mathrm{F}}, \mathrm{LC}) = 1$ and $C^\tau(s_t^{\mathrm{S}}, \mathrm{LC}) = 2$. Thus, the same observed

summary can correspond to different finite-horizon pivot costs, establishing pivot-cost ambiguity. The complete construction and verification are provided in Appendix E.1. □

4.2 Joint versus Marginal Uncertainty in Decision-Making

Conditioned on the available observations, all candidate-rule costs $C(a_1), \ldots, C(a_k)$ are determined by the same underlying simplex state and may therefore be statistically dependent. Independently sampling each cost from its marginal distribution can break this joint structure and distort the relative ordering of candidate rules.

Proposition 2 (Decision Inconsistency under Independent Sampling)**.** *Suppose cost predictions for different actions are sampled independently from their marginal distributions. Then the probability of selecting a suboptimal action can be strictly higher than under joint sampling via a shared latent variable. Here, suboptimality is defined with respect to expected cost.*

Proof sketch. Consider two actions with costs $C(a_1) = Z + 1$ and $C(a_2) = Z + 2$, where Z takes values 0 and 2 with equal probability. Action a_1 has lower expected cost. Joint sampling using the same Z always preserves $C(a_1) < C(a_2)$. Under independent marginal sampling, the sampled costs are $Z_1 + 1$ and $Z_2 + 2$, where Z_1, Z_2 are independent copies of Z. When $Z_1 = 2$ and $Z_2 = 0$, which occurs with probability $1/4$, the ordering is reversed and the suboptimal action a_2 is selected. Thus, the suboptimal-selection probability is $1/4$ under independent sampling and 0 under joint sampling. See Appendix E.2 for the full proof. □

4.3 Two-Dimensional Epistemic Uncertainty in EpiPivot

Propositions 1 and 2 illustrate how compact observations can omit cost-relevant information and how independent marginal sampling can distort action comparisons. These observations motivate addressing epistemic uncertainty along the *temporal* and *action* dimensions within a unified framework.

Temporal dimension. Proposition 1 illustrates that a current observation may not uniquely determine future pivot costs. We therefore supplement the current summary with recent observation history $\mathcal{H}_t$, using temporal attention to capture patterns in optimization dynamics. This history can provide information beyond the current observation, with its practical contribution evaluated experimentally.

Action dimension. Proposition 2 shows that independent marginal cost sampling can break joint dependence and distort relative comparisons. At each decision step, EpiPivot uses a shared latent index to jointly predict candidate-rule costs, evaluating all rules under the same latent hypothesis.

Unified modeling. Our cost predictor takes the form $\hat{C}^\tau = f_\theta(\psi_\theta(\mathcal{H}_t, a), z)$, where $z \sim p(z)$ and $\psi_\theta(\mathcal{H}_t, a)$ is the action-conditioned trajectory embedding. The temporal encoder extracts information from recent optimization history, while the shared z models joint uncertainty across candidate-rule costs at the current decision step. For a fixed history and sampled z, the model produces deterministic predictions for all candidate rules; varying z yields different joint cost predictions. At decision time, a single z is sampled and the rule is selected as $a_t = \arg\min_a f_\theta(\psi_\theta(\mathcal{H}_t, a), z)$. Thus, EpiPivot combines history-conditioned cost prediction with joint sampling across candidate rules, addressing both dimensions within a single decision mechanism.

4.4 Preservation of Classical Simplex Guarantees

At each iteration, EpiPivot selects a classical pivot rule to determine the entering variable. The leaving variable is determined by a consistently initialized lexicographic minimum-ratio test, providing an anti-cycling mechanism across rule changes.

Proposition 3 (Correctness)**.** *Assume exact arithmetic, a feasible initial basis, a finite LP optimal value, and a lexicographically feasible initialization for the lexicographic minimum-ratio test. Then EpiPivot preserves feasibility and terminates at an optimal basis after finitely many pivots.*

Proof sketch. Each selected rule chooses an entering variable with negative reduced cost, while the lexicographic minimum-ratio test preserves primal feasibility. Under the stated initialization, lexicographic pivoting prevents any basis from recurring, regardless of which candidate rule selects

the entering variable. Since there are finitely many bases, the algorithm terminates after finitely many pivots. The finite optimal value excludes an unbounded improving direction, so termination occurs at an optimal basis. The full proof is provided in Appendix E.3. □

5 Experiments

In this section, we systematically examine the key modeling assumptions underlying *EpiPivot*, in addition to evaluating its empirical performance: (i) the role of epistemic uncertainty in simplex pivot selection; (ii) whether trajectory-level information provides a meaningful edge over single tableau snapshots; (iii) the performance gains of uncertainty-aware modeling relative to deterministic baselines; and (iv) generalization across different problem scales and families of linear programs.

5.1 Experimental Setup and Benchmarks

We consider LP problems of the form

$$\max_x \; c^\top x \quad \text{s.t.} \quad Ax \le b, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. All instances are solved using the primal simplex method, where at each iteration a pivot rule is used to select the entering variable.

Data Generation. We train on a mixture of three LP types: standard random, sparse and degenerate, and highly structured instances with correlated constraints, sampled at proportions 50%, 25%, and 25% respectively to encourage robust generalization. All models are trained on 100 randomly generated instances yielding approximately 1.5 million training samples, and evaluated on 50 independent test instances per setting.

Pivot Rules and Benchmarks We compare against three classical pivot rules: the LC rule, the Steepest-edge rule, and Bland's rule, spanning a spectrum from aggressive to conservative behavior. These same rules form the action space of EpiPivot. We additionally include RLRule [Suriyanarayana et al., 2022], a reinforcement learning-based pivot-rule selector, as a learning-based baseline.

Evaluation Metrics. We evaluate all methods using *weighted pivot cost*, the sum of per-iteration costs over Phase-II pivots, where LC and Bland are assigned unit cost and Steepest-edge is assigned 1.15 per step to reflect its additional overhead [Suriyanarayana et al., 2022]. This metric is used consistently as both the training supervision signal and the test evaluation criterion.

5.2 Efficiency of Uncertainty-Aware Pivot Selection

We evaluate the proposed uncertainty-aware pivot selection strategy on held-out LP instances of the same scale as those used for training. This experiment assesses whether EpiPivot reduces weighted pivot cost relative to classical pivot rules and the learning-based baseline. Table 1 reports the average weighted pivot cost for *EpiPivot*, classical pivot rules, and a reinforcement-learning-based pivot-rule controller (RLRule) [Suriyanarayana et al., 2022].

Size	EpiPivot	RLRule	LC	Steepest	Bland
(25,50)	**18.71 ±4.91**	33.90 ±25.73	69.38 ±40.45	27.54 ±16.63	164.62 ±68.51
(50,75)	**43.35 ±18.10**	73.69 ±56.92	127.33 ±80.46	59.68 ±28.18	212.67 ±121.12
(100,100)	**67.58 ±21.22**	83.50 ±75.60	144.77 ±69.66	76.77 ±35.98	262.00 ±74.10

Table 1: Mean weighted pivot cost (mean ± 95% CI half-width).

Across all tested problem sizes, *EpiPivot* achieves the lowest weighted pivot cost. Relative to RLRule, it attains over 40% relative speedups in weighted pivot cost on certain problem sizes. This suggests that explicitly predicting future algorithmic cost can help capture the long-term impact of pivot decisions. Compared to classical pivot rules, it reduces weighted pivot cost by up to 73% relative to the LC rule and nearly 90% relative to Bland's rule. Although the Steepest rule is the strongest fixed baseline across problem sizes, *EpiPivot* consistently improves upon it, indicating that no single deterministic pivot heuristic is sufficient throughout the simplex trajectory.

From a modeling perspective, these gains arise from explicitly representing epistemic uncertainty in the remaining algorithmic cost. By maintaining distributions over finite-horizon costs and applying

Thompson sampling, *EpiPivot* evaluates candidate pivot rules under a shared sampled future, enabling consistent trade-offs between aggressive and conservative strategies. This uncertainty-aware decision mechanism leads to lower average pivot cost with controlled variability, as reflected by the 95% confidence intervals.

5.3 Ablation Studies

We perform ablation studies to empirically examine the key modeling assumptions underlying the proposed method, with a particular focus on the role of trajectory conditioning and the explicit representation of epistemic uncertainty in guiding pivot selection. All experiments in this subsection use LP instances of size $(25, 50)$ for both training and evaluation. All ablation results are averaged over five evaluation seeds, with 95% confidence intervals reported.

Effect of History Length H We first investigate the impact of the history length H used in the framework, which determines how many past pivot states are incorporated into the decision process. We vary H from 1 to 32 while keeping all other components fixed. Fig. 3a summarizes the results on the two classes of linear programming problems introduced in Section 5.1. Means and 95% confidence intervals over five evaluation seeds are reported in Appendix F.1. For standard random LPs, performance exhibits only mild sensitivity to the choice of H. This behavior is consistent with the fact that such instances typically converge within a small number of simplex iterations, leaving limited opportunity for extended historical context to influence pivot decisions. In contrast, more challenging LP instances, such as those with strong degeneracy, tend to induce much longer pivot trajectories, making pivot selection increasingly dependent on historical information. Overall, these results indicate that for more challenging LP instances, historical context can effectively compensate for the limitations of purely local decision signals.

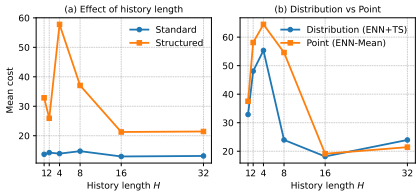


Figure 3: Ablation studies on trajectory information and epistemic uncertainty. (a) Effect of history length H on weighted pivot cost for standard vs. structured LP instances. (b) Comparison between distributional and point-prediction controllers under varying history lengths.

Point Prediction vs. Distributional Prediction. To isolate the role of epistemic uncertainty, we compare a point-prediction variant (ENN-Mean), which selects pivot rules based on posterior mean cost by averaging ENN predictions over multiple index samples, against the full distributional controller that explicitly represents epistemic uncertainty via Thompson sampling. Both variants share the same architecture and training data, differing only in how uncertainty is used at decision time. Fig. 3b shows that distributional decision making yields clear advantages when trajectory history is limited. These gains are most pronounced at intermediate history lengths, where trajectory information is informative while epistemic uncertainty remains substantial. In contrast, point prediction requires substantially longer history to achieve comparable performance. As the history length increases further, the performance gap narrows, consistent with diminishing posterior uncertainty. These results indicate that modeling epistemic uncertainty improves the efficiency of information use, enabling effective pivot selection with shorter histories. Detailed numerical comparisons are reported in Appendix F.2.

Shared vs. Independent Latent Sampling. To examine the effect of joint sampling across rules, we vary only the latent-sampling scheme at inference. Shared-z uses one latent sample for all candidate rules, whereas independent-z samples a separate latent index for each rule. Table 2 reports weighted pivot cost, success rate, and rule-selection agreement. Agreement denotes the percentage of paired decisions on identical inputs where the two schemes select the same rule.

Table 2: Shared vs. independent latent sampling.

Metric	Standard	Structured
Shared-z cost	12.73 ± 2.10	27.50 ± 2.60
Independent-z cost	13.07 ± 3.08	39.75 ± 5.30
Agreement	97.9%	47.7%
Shared-z success	100%	100%
Independent-z success	100%	85.2%

On standard LPs, the two schemes achieve similar costs and identical success rates. On structured LPs, independent sampling increases mean weighted pivot cost by approximately 44.5% and reduces success from 100% to 85.2%. These results support the practical value of preserving joint cost predictions across rules, particularly on structurally challenging instances.

Effect of Rollout Horizon τ. We investigate the sensitivity of EpiPivot to the rollout horizon τ, which controls the length of counterfactual rollouts used to generate training labels. We vary τ from 5 to 25 while keeping all other components fixed. Table 3 reports the weighted pivot cost on both standard and structured instances.

τ	Standard	Structured
5	15.00	31.30
10	14.48	32.55
15	**12.75**	**27.92**
25	12.99	28.97

Table 3: Effect of rollout horizon τ on weighted pivot cost.

Both instance types achieve the best performance at $\tau = 15$, and the variation across different τ values remains moderate, which we attribute to the 25-dimensional size-invariant feature representation: by capturing stable structural signals from the tableau, rollout costs remain within a consistent range across different τ values, reducing sensitivity to its exact choice. In particular, a large τ does not further improve the results and can even degrade the performance due to increased label variance.

Learned Switching vs. Heuristic Switching. To verify that performance gains arise from learned decision-making rather than switching frequency alone, we compare EpiPivot against two additional baselines: (1) **Random**, which uniformly selects among the three pivot rules at each iteration; and (2) **Stagnation**, an event-triggered heuristic that begins with LC, switches to Steepest after $K = 5$ consecutive iterations with insufficient objective improvement, and falls back to Bland after a further $M = 5$ stagnation steps.

EpiPivot outperforms the stagnation-based heuristic by 22% on standard instances and 55% on structured instances. More notably, on structured instances the stagnation heuristic achieves only 87.5% success rate and random switching only 75%, while EpiPivot maintains 100% success rate across all trials. These results confirm that the performance gains reflect genuine learned decision-making under epistemic uncertainty, rather than the effect of switching frequency alone.

Method	Standard	Success	Structured	Success
EpiPivot	**12.75**	100%	**27.95**	100%
Stagnation	16.38	100%	63.06	87.5%
Random	18.26	100%	83.66	75%

Table 4: Comparison against heuristic switching baselines (weighted pivot cost). A trial is successful if the simplex method reaches optimality within 300 iterations.

5.4 Generalization

5.4.1 zero-shot generalization

Beyond generic LP instances, we evaluate our method on problem families with distinct geometric and algorithmic characteristics. Across all experiments, we use a fixed history length $H = 4$, train on 100 instances, and evaluate on 40 held-out test instances. Problem formulations for transportation problems and LP relaxation of Traveling Salesman Problem (TSP), provided in Appendix A.3.

Transportation Problems. As shown in Table 5, on transportation instances of size 6×8 (with 6 suppliers and 8 demanders), our approach reduces weighted pivot cost by approximately 18.5%, 24.0%, and 39.4% relative to LC, Steepest, and Bland, respectively. The tight confidence intervals indicate stable behavior without unnecessary exploration.

LP Relaxations of TSP. The traveling salesman problem exhibits stronger path dependence and more complex pivot dynamics. On TSP-10 instances, our method reduces weighted pivot cost by approximately

Problem (size)	**Method**	**Cost (95% CI)**
Transport (6×8)	LC	8.40 [7.59,9.21]
	Steepest	9.01 [8.16,9.86]
	Bland	11.30 [9.82,12.78]
	EpiPivot	**6.85** [6.39,7.67]
TSP (10)	LC	89.12 [76.21,102.04]
	Steepest	47.97 [44.60,52.34]
	Bland	252.75 [221.79,283.71]
	EpiPivot	**40.55** [38.24,42.86]

Table 5: Generalization across LP families (mean cost, 95% CI).

54.5%, 15.5%, and 84.0% relative to LC, Steepest, and Bland, respectively, demonstrating the benefit of adaptive rule level control in challenging settings (see Table 5).

5.4.2 Netlib Benchmarks

We further evaluate *EpiPivot* on a collection of linear programs from the Netlib benchmark set, a standard testbed widely used to assess the practical performance of LP solvers [Gay, 1985]. These instances differ substantially from the synthetic training distribution in both scale and constraint structure.

Problem Normal	EpiPivot	LC	Steepest	Bland	Improv.†
sc105 (105×163)	**15.80**	21.00	19.55	19.00	**+16.8%**
sc205 (205×317)	**24.30**	43.00	27.60	42.00	**+12.0%**
kb2 (52×77)	**25.40**	43.00	27.60	64.00	**+8.0%**
adlittle (56×138)	**39.20**	58.00	42.55	97.00	**+7.9%**
×10X					
agg2 (516×758)	**35.30**	39.00	41.40	63.00	**+9.5%**
standmps (587×1394)	**168.35**	180.00	186.30	653.00	**+6.5%**
etamacro (617×1033)	**330.55**	587.00	348.45	801.00	**+5.1%**
grow7 (420×581)	**106.90**	112.00	117.30	295.00	**+4.6%**

Table 6: Netlib Phase-II performance measured by weighted pivot cost. †Improvement over the best classical baseline. Normal: $\max(m/40, n/80) < 10$; $\times$10X otherwise.

All models are trained using the same setup as in previous sections, with the training problem size fixed to 40×80. The selected instances span a wide range of problem sizes and structures, with problem dimensions exceeding the training scale by more than $10\times$ in both directions. Table 6 reports the Phase-II weighted pivot cost for each instance, along with per-instance comparisons against all classical pivot rules.

Across all test problems where classical rules attain the optimal objective, *EpiPivot* achieves a lower cost than the strongest fixed baseline. These results suggest that rule-level control provides a stable abstraction for learning-based guidance in the simplex method, particularly when extrapolating to problem sizes beyond the training distribution.

6 Discussion

The empirical results confirm that uncertainty-aware pivot selection consistently outperforms both classical and learning-based baselines across problem types and scales. Beyond the experimental results, we provide extended discussion on several aspects. First, regarding the **action space**, EpiPivot is open to extension and naturally accommodates additional pivot heuristics, though performance gains stem from how rules are selected rather than how many rules are used. Second, regarding **computational overhead**, ENN inference introduces additional decision costs, but on some large, structurally challenging instances, EpiPivot offsets this overhead by avoiding inefficient pivots, thereby reducing total solution time. Furthermore, our **feature importance** analysis reveals that critical state signals are strongly instance-dependent, offering empirical guidance for state representation design in learning-based simplex methods. Finally, these properties collectively position EpiPivot as a natural stepping stone for **broader impact on operations research**, with direct implications for branch-and-bound, column generation, and cutting plane methods. Extended analysis is provided in Appendix G.

7 Conclusion

We introduce *EpiPivot*, a framework that treats pivot-rule selection as a decision problem under epistemic uncertainty, enabling the simplex method to adaptively switch among complementary pivot strategies based on predicted future algorithmic cost. Evaluated across synthetic, structured, and benchmark LP instances, *EpiPivot* reduces weighted pivot cost by up to 73% over classical rules and over 40% over reinforcement learning baselines, while reaching optimality on all tested instances. Crucially, these gains are achieved without modifying the underlying simplex algorithm, and the learned policy generalizes robustly to unseen problem scales and structures. Together, these findings offer a principled and practically effective response to the challenge raised by Bengio et al. [2021]: how to integrate learning into combinatorial optimization while preserving feasibility, correctness, and robustness.

References

Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

R. G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977.

D. Chételat and A. Lodi. Continuous cutting plane algorithms in integer programming. *Operations Research Letters*, 51(4):439–445, 2023.

B. A. Cipra. The best of the 20th century: Editors name top 10 algorithms. *SIAM News*, 33(4), 2000.

G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

J. J. Forrest and D. Goldfarb. Steepest-edge simplex algorithms for linear programming. *Mathematical Programming*, 57(1-3):341–374, 1992.

M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi. Exact combinatorial optimization with graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

D. M. Gay. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society COAL Newsletter*, 13:10–12, 1985.

R. G. Jeroslow. The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Mathematics*, 4(4):367–377, 1973.

E. Khalil, P. Le Bodic, L. Song, G. Nemhauser, and B. Dilkina. Learning to branch in mixed integer programming. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

V. Klee and G. J. Minty. How good is the simplex algorithm. *Inequalities*, 3(3):159–175, 1972.

J. Kotary, F. Fioretto, P. Van Hentenryck, and B. Wilder. End-to-end constrained optimization learning: A survey. *arXiv preprint arXiv:2103.16378*, 2021.

T. Liu, S. Pu, D. Ge, and Y. Ye. Learning to pivot as a smart expert. In *Proceedings of the aaai conference on artificial intelligence*, volume 38, pages 8073–8081, 2024.

S. Ma, J. Lee, N. Serban, and S. Yang. Deep attention q-network for personalized treatment recommendation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 329–337. IEEE, 2023.

C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.

R. Nowak. Generalized binary search. In *2008 46th annual Allerton conference on communication, control, and computing*, pages 568–574. IEEE, 2008.

I. Osband, Z. Wen, S. M. Asghari, V. Dwaracherla, M. Ibrahimi, X. Lu, and B. Van Roy. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36:2795–2823, 2023.

A. Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

V. Suriyanarayana, O. Tavaslıoğlu, A. B. Patel, and A. J. Schaefer. Reinforcement learning of simplex pivot rules: a proof of concept. *Optimization Letters*, 16(8):2513–2525, 2022.

Y. Tang, S. Agrawal, and Y. Faenza. Reinforcement learning for integer programming: Learning to cut. In *International Conference on Machine Learning*, pages 9367–9376. PMLR, 2020.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

A Background

A.1 The Simplex Method

The simplex method [Dantzig, 1963] is an iterative algorithm for solving linear programs of the form:

$$\max_x \ c^\top x \quad \text{s.t.} \quad Ax \leq b, \tag{2}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. The algorithm operates over the vertices of the feasible polytope, each corresponding to a BFS. Starting from an initial BFS, the simplex method iteratively moves to an adjacent vertex by performing a pivot operation: selecting an entering variable and a leaving variable to update the current basis. This process continues until no improving direction exists, at which point the current BFS is optimal.

The choice of entering variable at each iteration is governed by a pivot rule. Classical pivot rules include:

- **Largest Coefficient (LC) rule** [Dantzig, 1963]: selects the variable with the most negative reduced cost, targeting the locally steepest descent direction.
- **Steepest-edge rule** [Forrest and Goldfarb, 1992]: selects the entering variable based on a normalized reduced-cost criterion that accounts for geometric scaling of the feasible region.
- **Bland's rule** [Bland, 1977]: selects the first admissible entering variable by index, guaranteeing finite termination by avoiding cycling.
- **Greatest Improvement rule** [Jeroslow, 1973]: selects the entering variable that yields the largest actual improvement in the objective value, requiring a full ratio test for every candidate variable and thus incurring higher per-step computational cost.

Each rule embodies a fixed decision preference and can be effective in certain settings. However, because a single rule is applied throughout the entire solution process, it cannot adapt to changes in problem structure or different stages of the algorithm, often leading to inefficient pivot sequences on challenging instances.

A.2 Thompson Sampling

Thompson sampling [Thompson, 1933] is a decision-making strategy under uncertainty. At each decision step, parameters θ are sampled from a posterior distribution $p(\theta \mid \mathcal{D})$, and the action that is optimal under the sampled parameters is selected:

$$a_t = \arg\max_a \ \mathbb{E}[r(a) \mid \theta]. \tag{3}$$

Unlike point-estimate decision rules that minimize expected cost and may induce overconfident decisions, Thompson sampling conditions each decision on a plausible realization of future behavior, ensuring that all candidate actions are evaluated under a shared sampled model and enabling internally consistent comparisons across actions.

A.3 Linear Programming Formulations of Benchmark Problems

A.3.1 Transportation Problem Formulation

The transportation problem is a linear programming problem that seeks to minimize the total transportation cost while satisfying supply and demand constraints.

We consider the standard balanced transportation formulation [Schrijver et al., 2003], given as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{4a}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} = s_i, \quad \forall i \in \{1,2,\ldots,m\} \quad \text{(supply constraint)} \tag{4b}$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad \forall j \in \{1,2,\ldots,n\} \quad \text{(demand constraint)} \tag{4c}$$

$$x_{ij} \geq 0, \quad \forall i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\} \quad \text{(non-negativity)} \tag{4d}$$

A.3.2 MTZ Relaxations of the Traveling Salesman Problem

The traveling salesman problem is a combinatorial optimization problem that seeks the shortest tour visiting each city exactly once and returning to the starting city. We consider the standard Miller-Tucker-Zemlin formulation of the TSP [Miller et al., 1960], given as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} d_{ij}x_{ij} \tag{5a}$$

subject to:

$$\sum_{j=1,j\neq i}^{n} x_{ij} = 1, \quad \forall i \in \{1,2,\ldots,n\} \quad \text{(departure constraint)} \tag{5b}$$

$$\sum_{i=1,i\neq j}^{n} x_{ij} = 1, \quad \forall j \in \{1,2,\ldots,n\} \quad \text{(arrival constraint)} \tag{5c}$$

$$u_i - u_j + n \cdot x_{ij} \leq n-1, \quad \forall i,j \in \{2,3,\ldots,n\}, \quad i \neq j \quad \text{(subtour elimination)} \tag{5d}$$

$$u_i \in \mathbb{Z}_+, \quad \forall i \in \{1,2,\ldots,n\} \tag{5e}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \neq j, \quad i,j \in \{1,2,\ldots,n\} \tag{5f}$$

In our experiments, we use the LP relaxation, replacing $x_{ij} \in \{0,1\}$ with $0 \leq x_{ij} \leq 1$ and relaxing u_i to continuous nonnegative variables.

B State Feature Definitions

We represent the algorithmic state of the simplex method using a **25-dimensional, size-invariant feature vector** extracted from the current simplex tableau. All features are computable in time linear in the tableau size and do not depend on the absolute problem dimensions (m, n). They are designed to summarize decision-relevant geometric, numerical, and algorithmic properties of the current basis.

B.1 Reduced Cost Features (7)

Let $r \in \mathbb{R}^{n+m}$ denote the reduced cost vector associated with the current simplex tableau, excluding the right-hand-side column.

1. **Number of improving directions.** The number of columns with strictly negative reduced cost:
$$\sum_j \mathbb{I}[r_j < 0].$$
2. **Fraction of improving directions.** The proportion of columns whose reduced cost is negative.

3. **Minimum reduced cost.** The smallest reduced cost value, corresponding to the locally most improving direction.
4. **Mean reduced cost.** The arithmetic mean of the reduced cost vector.
5. **Standard deviation of reduced costs.** A measure of dispersion among reduced cost values.
6. **Mean absolute reduced cost.** The mean of $|r_j|$, capturing the overall magnitude of reduced cost signals.
7. **Reduced cost gap.** The difference between the smallest and second smallest reduced costs, measuring how clearly separated the best local entering direction is.

B.2 Right-Hand-Side and Degeneracy Features (4)

Let $b \in \mathbb{R}^m$ denote the right-hand-side values of the tableau.

8. **Minimum right-hand-side value.** The smallest entry of the right-hand-side vector.
9. **Mean right-hand-side value.** The arithmetic mean of right-hand-side entries.
10. **Standard deviation of right-hand-side values.** The standard deviation of the right-hand-side vector.
11. **Fraction of degenerate basic variables.** The proportion of right-hand-side values that are close to zero, indicating degeneracy in the current basic solution.

B.3 Geometric Features of the Constraint Matrix (6)

Let $\bar{A} \in \mathbb{R}^{m \times (n+m)}$ denote the constraint coefficient block of the simplex tableau.

12. **Mean column norm.** The mean Euclidean norm of the columns of $\bar{A}$.
13. **Standard deviation of column norms.** The standard deviation of column norms.
14. **90th percentile of column norms.** The 90th percentile of the column norm distribution.
15. **Mean row norm.** The mean Euclidean norm of the rows of $\bar{A}$.
16. **Standard deviation of row norms.** The standard deviation of row norms.
17. **90th percentile of row norms.** The 90th percentile of the row norm distribution.

B.4 Optimization Context Features (2)

18. **Objective constant term.** The constant term of the objective function in the current tableau.
19. **Iteration count.** The number of simplex pivot steps performed so far.

B.5 Joint Rule-Aware and Feasibility Features (6)

The following features explicitly capture interactions between reduced costs, constraint geometry, and candidate entering variables. They are particularly informative for distinguishing pivot rules that normalize reduced costs by column geometry (e.g., steepest edge).

20. **Minimum normalized reduced cost.** The minimum value of $r_j / \|\bar{A}_{:,j}\|_2$ among improving directions.
21. **Mean normalized reduced cost.** The mean of $r_j / \|\bar{A}_{:,j}\|_2$ over improving directions.
22. **Normalized reduced cost gap.** The difference between the smallest and second smallest normalized reduced costs.
23. **Mean fraction of positive coefficients in candidate columns.** For a subset of improving columns, the average fraction of strictly positive entries in $\bar{A}_{:,j}$, serving as a proxy for feasibility and ratio-test stability.
24. **Correlation between $|r_j|$ and column norms.** The correlation between absolute reduced costs and column norms, measuring alignment between objective improvement and geometric scaling.

25. **Minimum improving column index.** The smallest column index among improving directions, serving as a proxy for Bland's rule selection behavior and helping distinguish index-based from cost-based pivot strategies.

C ENN Architecture and Configuration

Input Tokenization. At each step t, the input token for pivot rule r is constructed by concatenating the 25-dimensional state feature vector with a 3-dimensional one-hot encoding of r, yielding a 28-dimensional token. The one-hot encoding is injected only at the last time step of the history window; all preceding tokens use a zero vector in place of the rule encoding.

Trajectory Embedding. Each 28-dimensional token is projected to $d = 64$ dimensions via a linear layer with positional embeddings. The projected sequence is then processed by a single-layer multi-head self-attention module with 4 heads and key size 16, equipped with a padding mask to handle variable-length histories. The attended representations are aggregated via masked mean pooling over valid (non-padded) timesteps, yielding a trajectory embedding $\phi(\mathcal{H}_t, a) \in \mathbb{R}^{64}$.

Latent Index. The latent index $z \in \mathbb{R}^8$ is sampled from $\mathcal{N}(0, I_8)$ and concatenated with $\phi(\mathcal{H}_t, a)$, giving a 72-dimensional input to the MLP head. During training, $M = 8$ index samples are drawn per training example, and the Huber loss ($\delta = 10.0$) between predicted and observed costs is averaged across all samples and candidate rules. During inference, a single z is sampled and shared across all candidate pivot rules to ensure consistent comparisons under Thompson sampling.

MLP Head. The MLP head consists of two hidden layers of size 64 with ReLU activations, producing a scalar cost prediction $\hat{C}^\tau$. The prior distribution $p(z) = \mathcal{N}(0, I_8)$ is used both during training and inference.

Optimality Check. At each rollout step, the algorithm terminates when all reduced costs are nonnegative ($r_j \geq 0$ for all j). This is the standard termination criterion of the primal simplex method and is sufficient to certify optimality of a primal-feasible basis [Dantzig, 1963]. This check requires no additional learning and is performed at each step of the counterfactual rollout used to generate training labels.

Rollout Parameters and Failure Penalty. Training labels are generated via τ-step rollouts under each candidate pivot rule. A constant penalty $C_{\text{fail}} = 2\tau \cdot c_{\max}$ is assigned to rollouts where the pivot operation becomes numerically invalid, where $c_{\max} = 1.15$ is the maximum per-step cost among all rules. This ensures the penalty strictly exceeds the maximum possible cost of any valid τ-step rollout, i.e., $\tau \cdot c_{\max} < C_{\text{fail}}$.

D Algorithm Details

Algorithm 1 describes the offline training procedure, which learns a trajectory-conditioned cost model for each candidate pivot rule using precomputed supervised rollout labels. Algorithm 2 describes the online control procedure, which applies the learned model to select pivot rules via Thompson sampling at each simplex iteration. Together, they constitute the complete EpiPivot framework.

E Proofs

E.1 Proof of Proposition 1 (Pivot-Cost Ambiguity)

We establish the claim by a counterexample to the sufficiency of a tableau summary for determining finite horizon pivot costs. Consider the following two LP instances:

$$\text{Fast:} \quad \max_{x \geq 0} \; x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 2, \quad x_3 \leq 0,$$

$$\text{Slow:} \quad \max_{x \geq 0} \; x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 2, \quad x_1 + x_3 \leq 0.$$

Algorithm 1 Learning Pivot-Rule Costs

Input: LPs; rules $\mathcal{R}$; history H; horizon τ; prior $p(z)$; samples M; epochs E
Output: ENN params θ
Initialize θ
for $e = 1$ to E **do**
  Sample LPs; initialize simplex envs
  for each decision step **do**
    Keep recent successful history $\mathcal{H}$ (len $\leq H$)
    for each $r \in \mathcal{R}$ **do**
      Build $\mathcal{H}^{(r)}$; $x^{(r)} \leftarrow \mathrm{Pool}(\mathrm{Attn}(\mathcal{H}^{(r)}))$
      Rollout rule r for $\leq \tau$; set $y^{(r)} \leftarrow$ finite-horizon weighted pivot cost, or C_{fail} if a pivot fails
    end for
    Sample $z_{1:M} \sim p(z)$; predict $\hat{y}^{(r)}(z_m)$ for all r, m
    Update θ by Huber$(\hat{\mathbf{y}}, \mathbf{y})$
  end for
end for
return θ

Algorithm 2 Online Pivot-Rule Selection

Input: ENN f_θ; rules $\mathcal{R}$; history H; max iters $T_{\max}$
Output: $\{r_t\}$
Initialize simplex env; $\mathcal{H} \leftarrow \emptyset$
for $t = 1$ to $T_{\max}$ **do**
  if current solution is optimal **then**
    break
  end if
  Update history $\mathcal{H}$; keep last H states
  Sample $z \sim p(z)$ (TS)
  for each $r \in \mathcal{R}$ **do**
    Build tokens from $\mathcal{H}^{(r)}$ (pad to H, inject one-hot of r at last step)
    $\hat{y}^{(r)} \leftarrow f_\theta(\text{tokens}; z)$
  end for
  $r_t \leftarrow \arg\min_{r \in \mathcal{R}} \hat{y}^{(r)}$; pivot with r_t
  if pivot invalid **then**
    break
  end if
end for

Both instances have optimal objective value 2, and $x = (0, 2, 0)$ is an optimal solution for both. We consider simplex pivots from their slack bases without presolve, and treat optimal states as absorbing.

Introducing slack variables u_1, u_2 gives the initial dictionaries

$$\begin{aligned} \text{Fast:} \quad & u_1 = 2 - x_1 - x_2, \\ & u_2 = -x_3, \end{aligned} \qquad \begin{aligned} \text{Slow:} \quad & u_1 = 2 - x_1 - x_2, \\ & u_2 = -x_1 - x_3. \end{aligned}$$

Both initial basic feasible solutions satisfy $x = (0, 0, 0)$ and $(u_1, u_2) = (2, 0)$. Under the objective-row convention $v - x_1 - x_2 = 0$, the initial reduced costs of (x_1, x_2, x_3) are $(-1, -1, 0)$.

Define the simplified observation map $\tilde{\phi}(s) = \{j : r_j(s) < 0\}$, where $r_j(s)$ denotes the reduced cost of nonbasic variable j. The two complete initial states therefore satisfy

$$\tilde{\phi}(s_t^{\mathrm{F}}) = \tilde{\phi}(s_t^{\mathrm{S}}) = \{1, 2\}.$$

Apply the LC rule, which selects the variable with the most negative reduced cost, with smallest-index tie-breaking. Thus, x_1 enters first in both instances.

Fast instance. The minimum-ratio test selects u_1 to leave with step length 2. Substituting $x_1 = 2 - x_2 - u_1$ into the objective gives $v = 2 - u_1$. Hence, the first pivot reaches an optimal basis.

Slow instance. The minimum-ratio test gives ratios 2 and 0, so u_2 leaves with zero step length. After this degenerate pivot, the dictionary becomes

$$\begin{aligned} x_1 &= -x_3 - u_2, \\ u_1 &= 2 - x_2 + x_3 + u_2, \\ v &= x_2 - x_3 - u_2. \end{aligned}$$

The only nonbasic variable with negative reduced cost is now x_2. It enters with step length 2, and u_1 leaves. Substitution gives $v = 2 - u_1$, so the second pivot reaches an optimal basis.

Consequently, for any $\tau \geq 2$,

$$C^\tau(s_t^{\mathrm{F}}, \mathrm{LC}) = 1 \neq 2 = C^\tau(s_t^{\mathrm{S}}, \mathrm{LC}).$$

Thus, the same tableau summary can correspond to different finite-horizon pivot costs under the same deterministic pivot rule. This counterexample shows that a tableau summary need not uniquely determine these costs, establishing the proposition. The summary omits how improving columns interact with degenerate constraints.

To connect this construction to Definition 1, let S_t assign probability $1/2$ to each initial state. Conditioned on their shared observation, the cost takes values 1 and 2 with equal probability, yielding

$$\mathrm{Var}\Big(C^\tau(S_t, \mathrm{LC}) \mid \tilde{\phi}(S_t)\Big) = \frac{1}{4} > 0.$$

Conditioning on the complete state instead determines the cost exactly. Thus, the uncertainty arises from information omitted by the observation, rather than randomness in simplex transitions. □

E.2 Proof of Proposition 2 (Decision Inconsistency under Independent Sampling)

We establish the claim by an explicit construction. Condition on a fixed observation and consider two actions a_1, a_2 with joint costs

$$C(a_1) = Z + 1, \qquad C(a_2) = Z + 2,$$

where $\mathbb{P}(Z = 0) = \mathbb{P}(Z = 2) = 1/2$. Their expected costs satisfy

$$\mathbb{E}[C(a_1)] = 2 < 3 = \mathbb{E}[C(a_2)].$$

Thus, a_1 is the unique action minimizing expected cost.

Under joint sampling, a single realization of Z is shared by both actions. Since $C(a_2) - C(a_1) = 1$ for every realization, the sampled-cost minimizer is always a_1. The probability of selecting the suboptimal action is therefore

$$P_{\text{joint}} := \mathbb{P}(C(a_2) < C(a_1)) = 0.$$

Under independent marginal sampling, let Z_1, Z_2 be independent copies of Z and define

$$\widetilde{C}(a_1) = Z_1 + 1, \qquad \widetilde{C}(a_2) = Z_2 + 2.$$

Each sampled cost has exactly the same marginal distribution as its jointly sampled counterpart; only the dependence between actions changes. The suboptimal action a_2 is selected if and only if $Z_1 = 2$ and $Z_2 = 0$. There are no ties. Consequently,

$$\begin{aligned} P_{\text{ind}} &:= \mathbb{P}\Big(\widetilde{C}(a_2) < \widetilde{C}(a_1)\Big) \\ &= \mathbb{P}(Z_1 = 2, Z_2 = 0) = \frac{1}{4} > 0 = P_{\text{joint}}. \end{aligned}$$

This construction shows that independent marginal sampling can increase the probability of selecting an action with higher expected cost, even when all marginal distributions are preserved exactly. It establishes the claimed possibility, rather than a universal advantage of joint sampling for every cost distribution. □

E.3 Proof of Proposition 3 (Correctness)

We consider the standard-form LP under the assumptions of Proposition 3. Let B_0 denote the feasible initial basis. At each iteration, the selected rule chooses a nonbasic variable with negative reduced cost under our objective-row convention. The leaving variable is determined by the lexicographic minimum-ratio test defined below.

Lexicographic initialization. Let ε be a positive formal infinitesimal and define

$$e(\varepsilon) = (\varepsilon, \varepsilon^2, \ldots, \varepsilon^m)^\top, \qquad b(\varepsilon) = b + B_0 e(\varepsilon).$$

The perturbation is symbolic: it defines lexicographic comparisons without requiring a numerical value for ε or changing the original LP data. The same perturbation is maintained throughout the algorithm, regardless of changes in the selected rule.

At the initial basis,

$$B_0^{-1} b(\varepsilon) = B_0^{-1} b + e(\varepsilon).$$

Since $B_0^{-1} b \geq 0$, every initial basic variable is strictly positive in lexicographic order.

For a current basis B, write $\bar{b} = B^{-1} b$ and $M = B^{-1} B_0$. The perturbed basic solution is

$$x_B(\varepsilon) = \bar{b} + M e(\varepsilon).$$

Its ith component is represented by the coefficient vector $(\bar{b}_i, M_{i1}, \ldots, M_{im})$, with the constant term compared first.

For an entering variable x_j, let $d = B^{-1} A_j$. Among rows with $d_i > 0$, the leaving row minimizes

$$\left(\frac{\bar{b}_i}{d_i}, \frac{M_{i1}}{d_i}, \ldots, \frac{M_{im}}{d_i} \right)$$

lexicographically. Thus, when the ordinary minimum ratio is unique, the usual leaving row is selected. Only ties require comparison of the remaining coefficients. These coefficients are updated with every basis change, rather than reinitialized when a tie occurs. Since M is nonsingular, distinct rows cannot have identical normalized coefficient vectors, so the leaving row is unique.

Feasibility preservation. This rule is the minimum-ratio test for the symbolically perturbed problem. Starting from a strictly positive perturbed basic solution, the unique minimizing row gives a strictly positive perturbed step length. After the pivot, the entering variable and all remaining basic variables are strictly positive in lexicographic order. This property therefore holds inductively throughout the algorithm.

The constant terms of these perturbed basic variables are nonnegative. Consequently, every visited basis is feasible for the original LP. Equivalently, the lexicographic test always selects a minimizer of the ordinary ratio test and therefore preserves primal feasibility.

Absence of cycling. Under our sign convention, the entering variable has reduced cost $r_j = c_B^\top B^{-1} A_j - c_j < 0$. For the positive perturbed step length $\delta(\varepsilon)$, the objective increment satisfies

$$\Delta v(\varepsilon) = \big(c_j - c_B^\top B^{-1} A_j\big)\delta(\varepsilon) = -r_j \delta(\varepsilon) > 0$$

in lexicographic order. Thus, the perturbed objective strictly improves after every pivot, even when the original objective remains unchanged during a degenerate pivot.

For the fixed perturbation, each basis uniquely determines its basic solution and objective value. Returning to a previously visited basis would contradict strict objective improvement. Hence, no basis can recur. This argument requires only a negative reduced cost for the entering variable and is unaffected by switching among candidate rules.

Finite termination and optimality. There are finitely many possible bases, so the absence of repeated bases implies finite termination. If an entering variable with negative reduced cost had no positive component in d, it would define a feasible improving ray for the original LP, contradicting its finite optimal value. Therefore, an eligible leaving row exists whenever an improving entering variable exists.

The algorithm consequently terminates with no negative reduced costs. Together with primal feasibility, this is the simplex optimality condition under our sign convention. Hence, EpiPivot preserves feasibility and reaches an optimal basis after finitely many pivots. □

F Additional Ablation Results

F.1 Effect of History Length

Table 7 reports weighted pivot costs across five evaluation seeds. For each history length H, the same 10 standard and 10 structured LP instances are evaluated under each seed.

Table 7: Mean weighted pivot cost with 95% confidence interval half-widths across five evaluation seeds under different history lengths H.

H	**Standard**	**Structured**
1	13.55 ± 1.33	33.79 ± 20.28
2	14.08 ± 2.01	26.40 ± 20.12
4	13.99 ± 1.80	58.79 ± 33.20
8	14.15 ± 1.60	37.43 ± 7.85
16	13.08 ± 1.42	22.03 ± 6.71
32	13.11 ± 1.45	22.30 ± 4.67

F.2 Distributional vs. Point Prediction

Table 8 reports a detailed comparison between distributional and point-prediction controllers under varying history lengths H.

G Extended Discussion

Scope of the Action Space and Relation to Classical Pivot Heuristics: The action space in our framework consists of three widely used pivot rules: LC, Steepest-edge, and Bland's rule, which

Table 8: Average weighted pivot cost for distributional and point prediction under different history lengths H.

H	**ENN+TS (Distribution)**	**ENN-Mean (Point)**
1	32.86	37.52
2	48.08	58.11
4	55.34	64.47
8	23.97	54.60
16	18.17	19.12
32	23.95	21.43

together capture complementary trade-offs between efficiency, robustness, and numerical stability. Importantly, these rules are not treated as competing alternatives, but as distinct algorithmic behaviors that can be adaptively leveraged at different stages of the simplex trajectory. This choice is intended as a representative and practically motivated subset. The framework itself is not restricted to these rules, and can naturally adapt additional pivot heuristics that satisfy the same correctness requirements.

To directly assess the effect of a larger action space, we extended our framework to include the Greatest Improvement (GI) rule [Jeroslow, 1973], yielding a four-rule variant (EpiPivot-4R). As reported in Supplementary Table 9 and illustrated in Supplementary Fig. 5 and Fig. 7, while EpiPivot-4R achieves marginal gains on certain small-scale instances, it underperforms the three-rule variant on larger-scale instances and incurs substantially higher decision overhead. This indicates that performance gains do not stem from the breadth of the action space, but from uncertainty-aware decision-making over a representative set of complementary pivot behaviors.

Computational Overhead and Practical Considerations: The overall runtime consists of pivot time and decision time. EpiPivot operates at the level of pivot-rule selection, with each pivot executed according to the selected classical rule. The trade-off between decision overhead and pivot efficiency also arises among classical rules: LC scans candidate variables for the most negative reduced cost; Steepest computes a normalized score for each candidate; GI performs a full ratio test for every candidate; and Bland uses a simpler selection procedure but may require more pivots. ENN inference introduces additional decision overhead, so EpiPivot can be slower when classical rules already produce efficient pivot trajectories. Its practical benefit depends on problem structure and the resulting search trajectory, and cannot be determined by problem size alone. On some large, structurally challenging instances, EpiPivot offsets this overhead by reducing inefficient pivots, thereby shortening total solution time. Runtime breakdowns are provided in Table 9, with pivot times and decision-time ratios shown in Fig. 7 and Fig. 6, respectively.

In practice, we observe that long solution times under fixed pivot rules can be associated with two situations: prolonged stalling due to degeneracy, where many pivots yield no objective improvement, and long search trajectories despite continued objective improvement. The following two instances illustrate the potential benefits of EpiPivot in these situations.

For degeneracy-induced stalling, EpiPivot can reduce inefficient pivots through occasional rule changes. For example, on `degen3`, 94.2% of the first 1000 pivots under the fixed Steepest rule yield no objective improvement, with a longest zero-progress streak of 325 pivots and no repeated bases within this diagnostic window. EpiPivot selects Steepest in 1002 of its 1041 pivots, choosing LC or Bland in only the remaining 39, yet reduces the longest zero-progress streak to 154 pivots and the total pivot count from 1152 to 1041. These observations suggest that even when the same rule is used for most pivots, a small number of critical decisions may improve the subsequent trajectory. Adaptive rule selection can also provide practical benefits when fixed rules continue to improve the objective but follow long search trajectories. For example, during the first 1000 pivots on `fit2d`, LC and Steepest exhibit zero-progress rates of only 2.7% and 3.2%, respectively, with no repeated bases, yet neither reaches optimality within 3600 seconds. In contrast, EpiPivot reaches optimality in 3178.87 seconds. These results suggest that its benefits may arise both from mitigating degeneracy-induced stalling and from finding more efficient search trajectories, allowing it to offset additional decision overhead on some structurally challenging instances.

Feature Importance Analysis: To assess the contribution of individual state features to pivot selection quality, we conduct a feature masking analysis in which each feature dimension is zeroed

out individually, and the resulting change in weighted pivot cost relative to the full-feature baseline is measured. A positive percentage indicates that zeroing out the corresponding feature leads to an increase in weighted pivot cost, reflecting its contribution. We evaluate across two LP instance types: standard instances and structured instances, which exhibit strong degeneracy and path dependence.

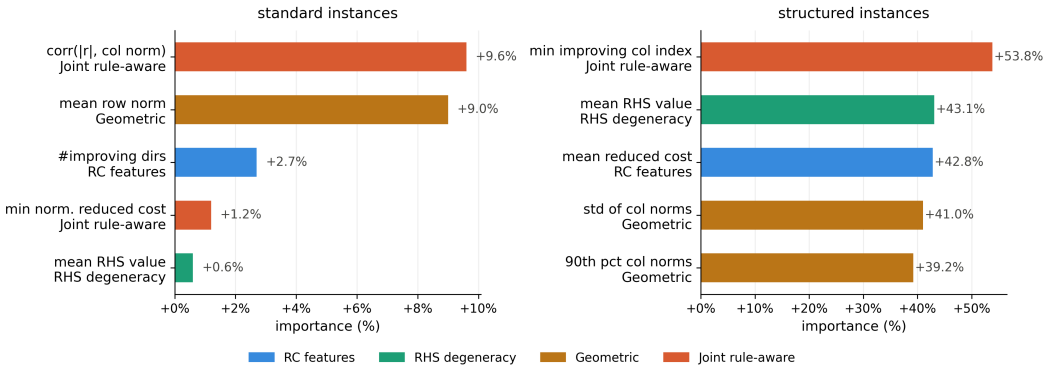


Figure 4: Top-5 most important features identified by feature masking analysis on standard and structured hard LP instances. The horizontal axis reports the percentage increase in weighted pivot cost upon zeroing out each feature, grouped by feature category. A larger value indicates a more critical feature for pivot selection quality.

The analysis reveals a notable asymmetry across instance types. In standard instances, a small subset of features dominates: the correlation between absolute reduced costs and column norms (Joint Rule-Aware, $+9.6\%$) and mean row norm (Geometric, $+9.0\%$) are the most critical. In contrast, structured instances tell a different story. As shown in Figure 4, critical features span multiple groups, with Joint Rule-Aware, RHS Degeneracy, and Geometric features all emerging as essential at substantially higher importance magnitudes. This directly reflects the increased role of path dependence and degeneracy in long-horizon pivot behavior, consistent with the epistemic uncertainty framework motivating EpiPivot.

To our knowledge, this is the first methodical empirical analysis of simplex state feature importance across LP instances of varying difficulty. This finding also provides empirical justification for the design of EpiPivot's state representation: by simultaneously covering reduced cost, degeneracy, geometric, and joint rule-aware feature groups, the 25-dimensional state vector allows the model to adaptively rely on the most relevant signals across instances of varying difficulty, achieving robust generalization without requiring instance-specific feature engineering.

Broader Impact on Operations Research. EpiPivot improves the simplex method as a foundational subroutine, opening several directions for broader impact in operations research. Since branch-and-bound for mixed-integer programming repeatedly solves LP relaxations, more efficient simplex pivoting directly reduces the total solving time at each node. Column generation methods rely on solving a sequence of LPs as pricing subproblems, where adaptive pivot selection could accelerate convergence of the master problem. Similarly, cutting plane methods invoke the simplex method after each cut addition, making pivot efficiency critical to overall performance. Extending EpiPivot to these settings represents a promising direction for learning-augmented combinatorial optimization. The feature importance analysis in Section G further provides empirical guidance for state representation design in these broader settings.

H Supplementary Experimental Results

H.1 Extended Netlib Benchmark Results

Table 9 reports the complete runtime breakdown across all Netlib benchmark instances, including both EpiPivot-3R and EpiPivot-4R variants, alongside all classical pivot rule baselines.

Table 9: Comparison of Pivot Methods on Netlib Benchmarks. Red = EpiPivot (4R), Blue = EpiPivot (3R), Green = GI (newly incorporated rule). **Bold** = lowest pivot cost (best) within each instance.

Problem	Size	Obj.	Method	Pivot Cost	Total (s)	Decision (s)	Pivot (s)	Decision (%)
Large-scale instances								
fit2d	(25×10500)	-6.85×10^4	EpiPivot (3R)	**22210.50**	3178.87	1434.40	1744.47	45.12%
			LC	/	3600+	/	/	/
			Steepest	/	3600+	/	/	/
			Bland	/	3600+	/	/	/
			GI	–	–	–	–	–
greenbeb	(2798×6008)	-4.30×10^6	EpiPivot (4R)	12021.9	731.750000	429.862400	301.887600	58.74%
			EpiPivot (3R)	**10051**	448.370000	206.337000	242.033000	46.02%
			LC	48710	1268.069000	540.742730	727.326270	42.64%
			Steepest	/	3600+	/	/	/
			Bland	/	3600+	/	/	/
			GI	/	3600+	/	/	/
nesm	(2433×4876)	1.41×10^7	EpiPivot (4R)	2878	77.871240	47.098930	30.772310	60.48%
			EpiPivot (3R)	**1873**	63.790100	37.809500	25.980600	59.27%
			LC	4371	74.680707	38.080897	36.599810	50.99%
			Steepest	2001	60.950331	33.132331	27.818000	54.36%
			Bland	/	1200+	/	/	/
			GI	3137.2	311.679751	150.323151	161.356600	48.23%
degen3	(1503×2604)	-9.87×10^2	EpiPivot (4R)	1288.5	29.46310	16.68212	12.7810	56.62%
			EpiPivot (3R)	**1191.3**	25.77370	14.510	11.27133	56.30%
			LC	/	600+	/	/	/
			Steepest	1324.8	28.923263	15.616753	13.30651	53.99%
			Bland	/	600+	/	/	/
			GI	2042.4	94.661	67.417	27.244	71.22%
Medium-scale instances								
standmps	(587×1394)	1.41×10^3	EpiPivot (4R)	170.65	3.213100	3.091367	0.121733	96.21%
			EpiPivot (3R)	**168.35**	1.087760	0.917747	0.170013	84.37%
			LC	180.00	0.258624	0.135453	0.123171	52.37%
			Steepest	186.30	0.501738	0.347505	0.154233	69.26%
			Bland	653.00	0.385563	0.191650	0.193913	49.71%
			GI	368.00	6.644304	6.549787	0.094517	98.58%
etamacro	(617×1033)	-7.56×10^2	EpiPivot (4R)	**319.00**	3.780100	3.478220	0.301880	92.01%
			EpiPivot (3R)	330.55	4.013000	3.663070	0.349930	91.28%
			LC	587.00	0.566303	0.205905	0.360398	36.36%
			Steepest	348.45	0.624580	0.372517	0.252063	59.64%
			Bland	801.00	2.389469	0.669661	1.719808	28.03%
			GI	356.50	3.528724	3.315034	0.213690	93.94%
agg2	(516×758)	-2.02×10^7	EpiPivot (4R)	**32.45**	1.137760	1.009857	0.127903	88.76%
			EpiPivot (3R)	35.30	1.179000	0.988802	0.190198	83.87%
			LC	39.00	0.298764	0.137337	0.161427	45.97%
			Steepest	41.40	0.205052	0.135336	0.069716	66.00%
			Bland	63.00	0.145460	0.067221	0.078239	46.21%
			GI	37.95	0.996484	0.926021	0.070463	92.93%
grow7	(420×581)	-4.78×10^7	EpiPivot (4R)	**102.75**	1.107640	1.034280	0.073360	93.38%
			EpiPivot (3R)	106.90	0.916320	0.854110	0.062210	93.21%
			LC	112.00	0.111624	0.030800	0.080824	27.59%
			Steepest	117.30	0.046779	0.024803	0.021976	53.02%
			Bland	295.00	0.667210	0.198151	0.469059	29.70%
			GI	103.50	0.484262	0.440693	0.043569	91.00%
Small-scale instances								
sc105	(105×163)	-5.22×10^1	EpiPivot (4R)	**13.90**	0.028781	0.025549	0.003232	88.77%
			EpiPivot (3R)	15.80	0.014322	0.010854	0.003468	75.79%
			LC	21.00	0.005498	0.001455	0.004043	26.46%
			Steepest	19.55	0.006224	0.002579	0.003645	41.44%

Continued on next page

Table 9 – continued

Problem	Size	Obj.	Method	Pivot Cost	Total (s)	Decision (s)	Pivot (s)	Decision%
			Bland	19.00	0.007658	0.001674	0.005984	21.86%
			GI	17.25	0.009977	0.004713	0.005264	47.24%
sc205	(205×317)	-5.22×10^1	EpiPivot (4R)	**24.30**	1.392800	1.342690	0.050110	96.40%
			EpiPivot (3R)	**24.30**	1.067890	1.010813	0.057077	94.66%
			LC	43.00	0.061575	0.020481	0.041094	33.26%
			Steepest	27.60	0.104442	0.040431	0.064011	38.71%
			Bland	42.00	0.091399	0.020464	0.070935	22.39%
			GI	28.75	0.119078	0.045874	0.073204	38.52%
kb2	(52×77)	-1.75×10^3	EpiPivot (4R)	27.40	0.534600	0.498490	0.036110	93.25%
			EpiPivot (3R)	**25.40**	0.334600	0.305300	0.029300	91.24%
			LC	43.00	0.063096	0.011690	0.051406	18.53%
			Steepest	27.60	0.049148	0.023764	0.025384	48.35%
			Bland	64.00	0.122617	0.042379	0.080238	34.56%
			GI	32.20	0.092362	0.056370	0.035992	61.03%
adlittle	(56×138)	2.25×10^5	EpiPivot (4R)	**26.30**	1.305300	1.270520	0.034780	97.34%
			EpiPivot (3R)	39.20	1.951600	1.871690	0.079910	95.91%
			LC	58.00	0.242791	0.120955	0.121836	49.82%
			Steepest	42.55	0.282963	0.196203	0.086760	69.34%
			Bland	97.00	0.186155	0.063548	0.122607	34.14%
			GI	27.60	0.243145	0.204758	0.038387	84.21%

H.2 Action Space Analysis

EpiPivot-4R achieves marginal gains on certain small-scale instances, but incurs higher pivot cost than the three-rule variant on larger-scale instances, indicating that performance gains arise from uncertainty-aware decision-making rather than action space breadth. Fig. 5 provides a detailed per-instance comparison across all selected Netlib benchmark instances ordered by problem size.

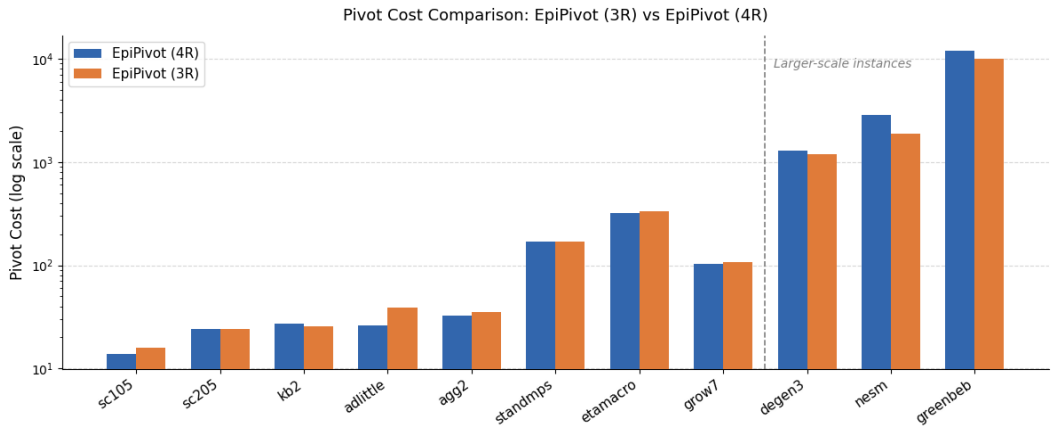


Figure 5: Pivot cost comparison between EpiPivot-3R and EpiPivot-4R across Netlib instances ordered by problem size.

H.3 Runtime Analysis

Fig. 6 shows the decision time ratio across Netlib instances ordered by problem size. The ratio is generally lower on larger instances.

Fig. 7 further compares the pivot time across all methods and instances.

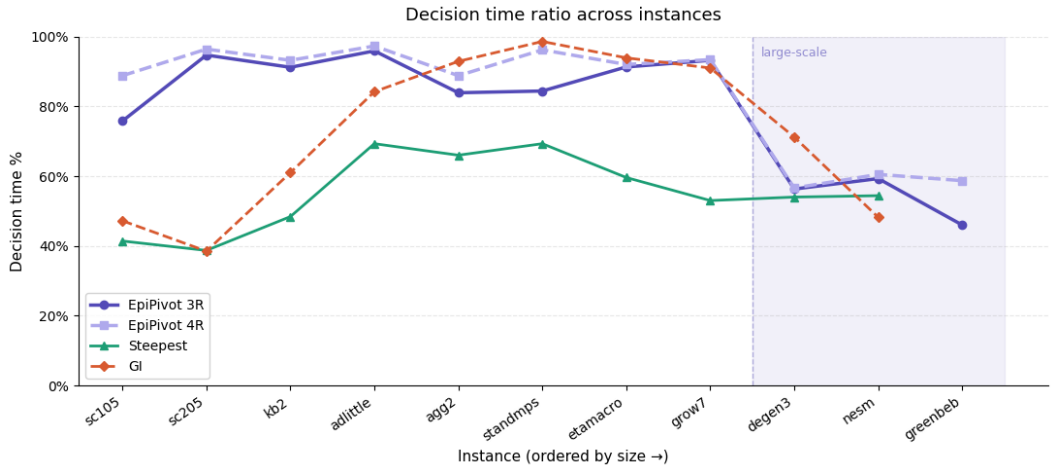


Figure 6: Decision time ratio across Netlib instances ordered by size. EpiPivot's decision overhead decreases as problem size grows, because pivot costs scale super-linearly while ENN inference operates on a fixed 25-dimensional representation.

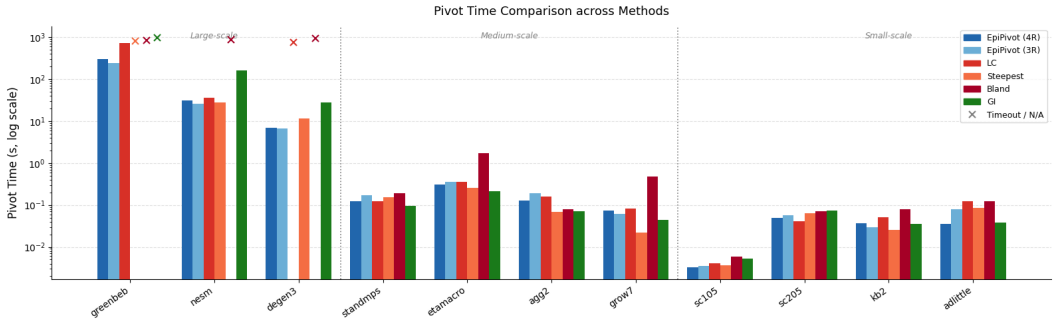


Figure 7: Pivot time comparison across methods and Netlib instances (log scale). EpiPivot achieves competitive pivot times across all scales, with classical rules failing to converge on large-scale instances.

I Impact Statement

This paper presents a methodological contribution to the field of mathematical optimization, specifically improving pivot-rule selection in the simplex method via uncertainty-aware decision-making. The proposed framework, *EpiPivot*, is a foundational algorithmic advancement with no direct negative societal impacts.

The primary potential benefits of this work lie in improving the computational efficiency of linear programming solvers, which are widely used in resource allocation, logistics, finance, and engineering applications. More efficient LP solvers can reduce computational costs and energy consumption in large scale optimization tasks. So we do not foresee any direct negative societal impacts arising from this work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that EpiPivot models epistemic uncertainty over future pivot costs and applies Thompson sampling for adaptive rule selection. These claims are directly supported by the theoretical analysis in Section 4, the experimental results in Section 5.2, and the ablation studies in Section 5.3. Additional supporting evidence, including extended benchmark results and implementation details, is provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the proposed approach are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results and their assumptions are stated in Section 4. Complete proofs are provided in the appendices.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental setup is described in Section 5.1, with additional implementation details in the appendices.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   The benchmark data are publicly available. An earlier version of the code was made available during review. The updated implementation is being cleaned up and documented for public release.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details, including data splits, and LP generation procedures, are described in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experimental results are reported with 95% confidence intervals, as shown in Tables 1, 5, and the ablation study tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on a standard CPU machine without requiring specialized hardware. The computational cost is moderate, and the experiments can be reproduced on a typical workstation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, this study fully complies with the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: An Impact Statement is included in the Appendix, discussing potential benefits of more efficient LP solvers and confirming no negative societal impacts are anticipated.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this paper are properly cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are released with the current version of the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve any crowdsourcing experiments or research with human subjects. All experiments are conducted on synthetically generated and publicly available LP instances.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve any research with human subjects, and no IRB approval or equivalent review was required.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Large language models are not used as any important, original, or non-standard component of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.