ANYTIME PREDICT-AND-OPTIMIZE WITH SOFT PRIMAL GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

In contextual combinatorial optimization, decision cost is a primary concern, but computation time can also be important, particularly in applications with operational deadlines. Conventional predict-and-optimize formulations focus on coefficient accuracy or terminal decision quality, but do not explicitly account for how prediction overhead and incomplete search jointly affect decisions before a deadline. We propose Soft Primal Prior (SPP), which combines coefficient regression with solution-membership supervision from cached feasible decisions. The predicted memberships define variable-level preferences that shape a linear surrogate objective and its feasible initialization, while an unmodified solver enforces the original constraints. This construction leaves every feasible decision admissible and uses one trained model across deadlines. After offline label construction, gradient updates require no optimizer calls. Our analysis relates budgeted regret to objective mismatch, membership-estimation error, and incomplete optimization, and identifies when later decision gains compensate for guidance latency. Across three contextual benchmarks and a retrospective energy-scheduling case, SPP achieves competitive budgeted decision quality and reduces end-to-end Area Under the Regret Curve (AURC) by up to approximately 19.3% relative to coefficient-only predict-then-optimize, while the energy case shows that early fallback costs can offset gains at later deadlines. The code to reproduce our experiments is available here.

1 INTRODUCTION

Many operational decisions depend on quantities that are not yet observed, such as future prices in energy scheduling or returns in resource allocation. Contextual optimization addresses this uncertainty through a predict-then-optimize pipeline: a model estimates unknown objective coefficients from observable features, and an optimizer selects a feasible decision (Donti et al., 2017; Elmachtoub & Grigas, 2022). Prediction error and decision cost need not rank models in the same order. Decision-focused learning therefore trains predictors according to their downstream decisions rather than coefficient error alone (Elmachtoub & Grigas, 2022; Mandi et al., 2022; Gupta & Huang, 2024).

Decision cost is a central concern in these systems, but computation time can also become a constraint when applications impose operational deadlines. Under such deadlines, a good terminal solution may not be available: prediction and model preparation consume part of the budget, and the solver may return an intermediate feasible solution. A learning objective defined through the optimum of the predicted problem does not capture this effect. The relevant criterion is the quality of the decision that the complete pipeline can deliver within the available time.

Existing work addresses computational efficiency through several routes. Surrogate-loss methods reduce the need for repeated optimization during training (Zharmagambetov et al., 2023; Berden et al., 2025), while learned primal heuristics and solution-guided search improve the quality–time trade-off for a supplied optimization problem (Paulus & Krause, 2023; Huang et al., 2024; Liu et al., 2025). These advances motivate a complementary question when the objective itself is unknown: *how should contextual predictions guide optimization when prediction and search share the same online budget?* We study this question through *anytime predict-and-optimize*, where true decision regret is evaluated across end-to-end time budgets rather than only after a complete solve (Zilberstein, 1996).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We propose Soft Primal Prior (SPP) to incorporate historical decision information into this budgeted pipeline. A dual-head model learns coefficient estimates and variable-membership preferences from cached feasible solutions. These two targets provide different information. Coefficients describe individual costs, whereas solution labels reflect selections under the instance's constraints (Khalil et al., 2022). At deployment, transformed membership scores supplement the coefficient estimates in a linear surrogate objective. The same objective guides feasible initialization and subsequent search by an unmodified solver. The scores express preferences rather than required assignments, so the original feasible set remains unchanged. One trained model serves all deadlines, and gradient updates require no optimizer calls after offline label construction. Our contributions are threefold:

- **An end-to-end anytime formulation of contextual optimization.** We formulate *anytime predict-and-optimize* for contextual combinatorial problems with unknown objective coefficients and limited online computation. The formulation replaces the exact optimization map with a deadline-dependent feasible decision and defines its objective through expected true regret across computation budgets. Prediction, guidance, and optimization share the same deadline, so decision quality depends on both objective estimation and the time available for search.
- **Soft primal guidance from offline decision supervision.** We propose **SPP**, which distills high-quality historical solutions into variable-level preferences that complement contextual coefficient estimates. These preferences shape a linear surrogate objective and guide feasible initialization before budgeted search. The method transfers decision information to an unmodified solver without hard variable fixing or changes to the feasible set. After offline label construction, gradient training requires no optimizer calls.
- **A theoretical account of the quality–time trade-off.** We derive a finite-budget regret bound that separates objective mismatch from residual optimization error at the deadline. Under an explicit decision-availability model, we also characterize when later quality improvements compensate for guidance latency. Together, these results explain why a better terminal decision does not necessarily yield a better end-to-end anytime system.

2 Related Work

Contextual optimization and decision-focused learning. Decision-focused learning trains predictors for downstream decisions rather than coefficient accuracy alone (Donti et al., 2017; Elmachtoub & Grigas, 2022). Differentiable convex layers, black-box differentiation, and randomized perturbations provide gradient mechanisms for this purpose (Amos & Kolter, 2017; Agrawal et al., 2019; Vlastelica et al., 2020; Berthet et al., 2020). Alternative objectives use feasible-solution rankings or directional perturbations, and recent theory addresses model misspecification (Mandi et al., 2022; Gupta & Huang, 2024; Bennouna et al., 2025). Computationally efficient training is a related concern. Problem relaxations and solution caches reduce oracle costs (Mandi et al., 2020; Mulamba et al., 2021); learned loss models approximate decision loss (Shah et al., 2022; Zharmagambetov et al., 2023); and LAVA uses precomputed adjacent vertices to avoid solver calls during gradient updates (Berden et al., 2025). These approaches improve decision-aware prediction and its training efficiency. We address a complementary question: which decisions can the complete prediction–optimization pipeline deliver within an online deadline? Training efficiency alone does not determine this budgeted decision quality.

Decision supervision and solver guidance. Historical solutions can supervise objective models as well as decision predictors. Optimality-based supervision connects contextual and inverse optimization (Sun et al., 2023), while task-informed metrics and prediction-bias correction provide other routes to decision-aware models (Bansal et al., 2023; Yang et al., 2025). SurCo learns linear surrogate objectives for nonlinear combinatorial tasks (Ferber et al., 2023). On the solver side, MIP-GNN learns variable biases from near-optimal solutions for node selection and warm starts (Khalil et al., 2022). ConPaS uses contrastive solution prediction to guide restricted search (Huang et al., 2024), and Apollo-MILP combines prediction and correction with uncertainty-based variable fixing (Liu et al., 2025). Our method combines coefficient supervision with solution-membership supervision, then uses the predicted preferences to shape the linear objective and feasible initialization. The distinction lies in the use of solution information: it supplements an estimated objective without hard

assignments or restrictions on the original feasible set. This construction retains a standard solver for the current instance's combinatorial constraints.

Anytime optimization. Anytime algorithms relate solution quality to available computation (Zilberstein, 1996), and the primal integral summarizes incumbent quality over a solver run (Berthold, 2013). Learned branching and cut selection improve solver efficiency (Gasse et al., 2019; Tang et al., 2020), while heuristic schedules (Chmiela et al., 2021), diving, and large-neighborhood search target the timely discovery of high-quality solutions (Paulus & Krause, 2023; Huang et al., 2023). Our formulation brings this quality–time perspective to contextual optimization with unknown objective coefficients. It evaluates regret under the realized objective at end-to-end deadlines, so prediction and guidance consume time that would otherwise be available for search. We summarize this profile through Area Under the Regret Curve (AURC) and use soft primal guidance to improve the decisions delivered within budget. The focus is therefore neither terminal decision quality nor the speed of solving an unchanged predicted problem in isolation, but the interaction between objective estimation, learned guidance, and incomplete optimization.

3 PRELIMINARIES

3.1 CONTEXTUAL COMBINATORIAL OPTIMIZATION

We consider decisions whose objective coefficients are unknown when the decision must be made. Each instance provides observable context x and known constraints ξ, while the realized coefficients $c \in \mathbb{R}^n$ are unavailable at deployment. Using the minimization convention, we define

$$\mathcal{Z}(\xi) = \{z \in \{0,1\}^n : A_\xi z \le b_\xi\}, \qquad z^\star(c,\xi) \in \arg\min_{z \in \mathcal{Z}(\xi)} c^\top z. \tag{1}$$

We assume $\mathcal{Z}(\xi)$ is nonempty. Profit maximization is represented by negated coefficients. The hindsight optimum $z^\star(c,\xi)$ uses the realized coefficients and provides the reference value $v^\star(c,\xi) = c^\top z^\star(c,\xi)$ for evaluation. Since c is unknown at deployment, a predictor $\widehat{c} = f_\theta(x,\xi)$ learns from historical data $\mathcal{D}_{\mathrm{tr}} = \{(x_i,\xi_i,c_i)\}_{i=1}^N$. A standard predict-then-optimize pipeline then solves the problem under $\widehat{c}$ (Elmachtoub & Grigas, 2022). With limited computation, however, the exact predicted optimum $z^\star(\widehat{c},\xi)$ may not be available in time. Our focus is therefore the feasible decision that the complete pipeline can return before a deadline. The test coefficients c remain hidden from the pipeline and are used only to evaluate its output.

3.2 END-TO-END BUDGETED DECISIONS

Let $\mathcal{A}$ denote the complete online pipeline. From (x,ξ), it constructs a solver objective $d_{\mathcal{A}}$ and, optionally, a feasible start. The deployed objective may be the coefficient prediction itself or incorporate additional learned guidance. We write $z_t^{\mathcal{A}} = \mathcal{A}_t(x,\xi) \in \mathcal{Z}(\xi)$ for the decision available at deadline t. The deadline covers all instance-specific online computation, not just the solver call. A new solver-backed candidate is credited only if its end-to-end latency satisfies $T_{\mathrm{E2E}} = T_{\mathrm{prep}} + T_{\mathrm{solve}} + T_{\mathrm{post}} \le t$, where preparation includes feature construction, prediction, guidance, feasible initialization, and model construction. Postprocessing includes solution extraction, feasibility verification, and return. Thus preparation and postprocessing reduce the time available for solver search. If no new verified candidate is ready by t, the pipeline returns a registered feasible fallback. We assume this fallback is available at each evaluated deadline and include any instance-specific cost of obtaining it in the online budget. These rules define decision availability rather than an operating-system hard real-time guarantee. Offline training and label construction are accounted for separately. One trained model and deployment configuration serve all deadlines. For comparison, a node limit b bounds solver search and yields $z_b^{\mathcal{A}}$, but does not constrain the surrounding online work.

3.3 ANYTIME DECISION QUALITY

At a given deadline, we evaluate the true cost of the available decision relative to the hindsight optimum. Its normalized regret is

$$r_{\mathcal{A}}(t;x,\xi,c) = \frac{c^\top z_t^{\mathcal{A}} - v^\star(c,\xi)}{Q(c,\xi)}, \tag{2}$$

where $Q(c, \xi) = |v^\star(c, \xi)| + \varepsilon$ with $\varepsilon > 0$. This measures decision quality under the realized coefficients c, rather than optimization progress under the solver objective $d_{\mathcal{A}}$. A single deadline does not describe performance over a range of computation budgets (Zilberstein, 1996). We therefore summarize the regret profile on $0 < t_{\min} < t_{\max}$ through the normalized Area Under the Regret Curve (AURC):

$$\mathrm{AURC}_{\mathcal{A}}(x, \xi, c) = \frac{1}{L} \int_{t_{\min}}^{t_{\max}} r_{\mathcal{A}}(t; x, \xi, c) \, \frac{\mathrm{d}t}{t}, \tag{3}$$

where $L = \log(t_{\max}/t_{\min})$. Following integral quality–time evaluation (Berthold, 2013), this criterion averages regret on a logarithmic time scale: equal multiplicative time intervals receive equal weight. Lower AURC indicates better average decision quality over the chosen budget range.

Our system-level objective is to minimize $\mathcal{J}(\mathcal{A}) = \mathbb{E}_{(x,\xi,c)\sim P}[\mathrm{AURC}_{\mathcal{A}}(x, \xi, c)]$ over pipelines that satisfy the information and deadline constraints, where P is the instance distribution. This is a design criterion for the complete pipeline; the supervised loss in Section 4 does not optimize AURC directly. Experiments observe a finite set of deadlines rather than the entire continuous profile. For $t_1 = t_{\min} < \cdots < t_K = t_{\max}$, we use trapezoidal integration in log time:

$$\widehat{\mathrm{AURC}}_{\mathcal{A}} = \frac{1}{\log(t_K/t_1)} \sum_{k=1}^{K-1} \frac{r_{\mathcal{A}}(t_k) + r_{\mathcal{A}}(t_{k+1})}{2} \log \frac{t_{k+1}}{t_k}, \tag{4}$$

where instance arguments are suppressed. All methods use the same budget grid within each task. We retain the actual output at every deadline: progress under $d_{\mathcal{A}}$ need not reduce regret under c. The estimate summarizes measured checkpoints but does not recover unobserved transitions between them. Reference construction and statistical aggregation are given in Appendix D.

4 Methodology

SPP has an offline learning stage and an online decision stage. Offline, a shared model learns coefficient estimates and membership preferences from historical coefficients and cached feasible solutions. Online, these outputs define a surrogate objective that guides both feasible initialization and solver search. An unmodified solver retains the original constraints, while the end-to-end deadline determines which decision is available for return.

4.1 Coefficient and Primal Supervision

The two supervision targets describe different aspects of an optimization instance. A coefficient label specifies a variable's contribution to the objective, whereas a membership label records whether that variable is selected in a feasible solution under the instance's constraints. For each labeled training instance $i \in \mathcal{I}_{\mathrm{lab}}$, we cache a feasible decision $\tilde{z}_i \in \mathcal{Z}(\xi_i)$ obtained under c_i and set $y_{ij} = \tilde{z}_{ij}$. This decision may be a certified optimum or a solver-limited incumbent. Both prediction heads use a shared MLP representation $h_i = H_\psi(x_i, \xi_i)$. The coefficient head produces $u_i = F_{\theta_c}(h_i)$, with normalized training target $\bar{c}_i = \mathcal{C}(c_i)$. The primal head produces logits $a_i = F_{\theta_p}(h_i)$ and membership probabilities $p_i = \sigma(a_i)$, where σ is the sigmoid function. Each head produces one output per decision variable. We train the shared representation and both heads jointly:

$$\begin{aligned} \mathcal{L} &= \beta \mathcal{L}_{\mathrm{coef}} + \alpha \mathcal{L}_{\mathrm{primal}}, \qquad \alpha, \beta > 0, \\ \mathcal{L}_{\mathrm{coef}} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\|u_i - \bar{c}_i\|_2^2}{n_i}, \\ \mathcal{L}_{\mathrm{primal}} &= \frac{1}{|\mathcal{I}_{\mathrm{lab}}|} \sum_{i \in \mathcal{I}_{\mathrm{lab}}} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell_{\mathrm{BCE}}(a_{ij}, y_{ij}), \end{aligned} \tag{5}$$

where $\ell_{\mathrm{BCE}}(a, y) = \mathrm{softplus}(a) - ya$ and n_i is the number of decision variables in instance i. The regression term fits the normalized coefficients, while the BCE term learns selection preferences from the cached decisions. The factor $1/n_i$ gives each instance an average per-variable loss. The primal loss targets marginal membership probabilities under the label-generation procedure, not a joint distribution over feasible solutions. A cached label represents one feasible selection and need not be the unique high-quality solution. Once these labels are cached, gradient updates require no solver calls. Architecture and training settings are given in Appendix C.3.

4.2 Soft Primal Objective Construction

SPP uses the predicted memberships to adjust objective costs rather than convert them into required binary assignments. Let $s = g(p)$ be the transformed membership scores. We combine them with the coefficient output through

$$\tilde{d}_\gamma = (1-\gamma)u - \gamma s, \qquad d_\gamma = \mathcal{N}(\tilde{d}_\gamma), \qquad 0 \le \gamma < 1. \tag{6}$$

The first term carries the predicted cost information. The second lowers the unscaled cost of selecting variables with larger membership scores. Thus γ controls the strength of online guidance, whereas α and β control training supervision. The normalization map only rescales the fused objective: $\mathcal{N}(v) = v/a(v)$ for $v \neq 0$, with $a(v) > 0$, and $\mathcal{N}(0) = 0$. Positive rescaling preserves exact minimizers, although it may affect finite-budget solver behavior. The score transformation, normalization, and fusion weight are fixed before test evaluation and remain unchanged across deadlines. This objective adjustment admits a soft-prior interpretation. Let $\lambda = \gamma/(1-\gamma)$ and define $q_s(z \mid x, \xi) \propto \exp(s^\top z)$ over $\mathcal{Z}(\xi)$. This distribution assigns greater weight to feasible decisions with larger total membership scores. The deployed problem is equivalent to

$$\arg\min_{z \in \mathcal{Z}(\xi)} d_\gamma^\top z = \arg\min_{z \in \mathcal{Z}(\xi)} \big[u^\top z - \lambda \log q_s(z \mid x, \xi)\big]. \tag{7}$$

The prior therefore expresses a preference among feasible solutions, not a restriction on which solutions the solver may choose. Its partition function is constant with respect to z and is never computed. This representation does not imply that BCE fits q_s or that p equals its constrained marginals.

To see how the adjustment can help, consider two feasible decisions z^+ and z^-, where z^+ has lower true cost. Write $\Delta z = z^+ - z^-$. If the coefficient head incorrectly prefers z^-, then $u^\top \Delta z > 0$. Guidance reverses this preference whenever $\lambda s^\top \Delta z > u^\top \Delta z$. Membership information can therefore correct a decision ordering without changing u or removing any feasible alternative. At $\gamma = 0$, deployment uses only the jointly trained coefficient head, which need not match a separately trained PTO predictor.

4.3 Feasible Initialization and Budgeted Deployment

A useful objective must also lead to a feasible decision that is available in time. For the evaluated tasks, SPP constructs a feasible start $z^0 = H_\xi(d_\gamma) \in \mathcal{Z}(\xi)$ using a deterministic task-specific procedure. We pass this start, the shaped objective d_γ, and the original constraints to an unmodified solver. Initialization supplies an incumbent, while the objective determines the preferences throughout the subsequent search. The deployment ablations vary these two mechanisms separately. Preparation consumes part of the same deadline as optimization. Let τ_{prep} be the elapsed time after feature processing, prediction, objective construction, initialization, and model building. Search and all operations needed to return a verified result must share the remaining $[t - \tau_{\text{prep}}]_+$ seconds. For a solver candidate $\hat{z}_t$, let $\tau_{\text{ready}}(\hat{z}_t)$ denote its end-to-end availability time, including extraction, verification, and return handling. The deadline output is

$$z_t = \begin{cases} \hat{z}_t, & \hat{z}_t \in \mathcal{Z}(\xi) \text{ and } \tau_{\text{ready}}(\hat{z}_t) \le t, \\ z_{\text{fb}}, & \text{otherwise}, \end{cases} \tag{8}$$

where z_{fb} is the registered feasible fallback from Section 3.2. The fallback need not coincide with z^0: the start initializes solver search, whereas the fallback provides an available decision when a new verified result is not ready. Candidate selection uses the surrogate objective, never the hidden coefficients c. One trained model and fusion rule serve all deadlines; only the available computation changes with t. Algorithm 2 and Appendix C.5 give the deployment workflow and task-specific feasibility conditions.

4.4 Quality–Time Analysis

SPP changes both the objective supplied to the solver and the computation available before a deadline. We analyze these effects through objective mismatch, membership-estimation error, and guidance latency. Fix an instance and suppress its arguments below. Write $d = d_\gamma$ and $Q = Q(c, \xi)$. A decision can incur true regret for two reasons: the deployed objective d may prefer different decisions from the true objective c, and the solver may not finish optimizing d before the deadline. To

distinguish these effects, define $G_t(a) = a^\top z_t - v^\star(a, \xi)$ for any objective vector a. This is the absolute suboptimality of the returned decision under a. Also define $\Delta_\xi(a) = \max_{z,z' \in \mathcal{Z}(\xi)} a^\top (z - z')$, which measures how much a varies across feasible decisions.

Proposition 1 (Budgeted regret). For every feasible deadline-available z_t and every $\kappa > 0$,

$$r_{\mathcal{A}}(t) \le \frac{\Delta_\xi(c - \kappa d) + \kappa G_t(d)}{Q}. \tag{9}$$

The first term measures objective mismatch over feasible decisions; the second measures incomplete optimization under the deployed objective. The analytical scale κ matches objective units and is not a deployment parameter. An exact solve under d removes the second term, but the mismatch term may remain. SPP can affect both terms: objective shaping changes decision preferences, while initialization and online overhead affect the residual reached by the deadline. For fixed d and κ across deadlines, averaging the same decomposition gives

$$\mathrm{AURC}_{\mathcal{A}} \le \frac{\Delta_\xi(c - \kappa d) + \kappa \overline{G}(d)}{Q}, \tag{10}$$

where $\overline{G}(d) = L^{-1} \int_{t_{\min}}^{t_{\max}} G_t(d)\, \mathrm{d}t/t$ is the log-time average residual.

We next isolate the effect of errors in the predicted memberships. Let $\eta_j = \mathbb{E}[Y_j \mid x, \xi]$ be the target membership probability under the label-generation procedure. Holding the coefficient output u fixed, compare the predicted-score objective $h_p = u - \lambda g(p)$ with the reference objective $h_\eta = u - \lambda g(\eta)$. The latter uses the target marginals rather than their predictions; it is an analytical reference, not the hidden true objective c. For $p \in (0,1)^n$, define the conditional excess BCE as $\mathcal{E}_{\mathrm{BCE}}(p; \eta) = n^{-1} \sum_{j=1}^n \mathrm{KL}\big(\mathrm{Bern}(\eta_j) \| \mathrm{Bern}(p_j)\big)$. This is the conditional BCE risk above its minimum at η.

Proposition 2 (Membership-estimation error). Suppose the complete score transformation g satisfies $\|g(p) - g(q)\|_1 \le L_g \|p - q\|_1$ for all $p, q \in [0,1]^n$. For every feasible returned decision,

$$0 \le h_\eta^\top z_t - v^\star(h_\eta, \xi) \le G_t(h_p) + \lambda L_g n \sqrt{\mathcal{E}_{\mathrm{BCE}}(p; \eta)/2}. \tag{11}$$

Thus suboptimality under the reference objective is bounded by the residual under the predicted-score objective and an additional membership-estimation term. For the same returned z_t, $G_t(h_p)$ is a positively rescaled version of $G_t(d_\gamma)$; the scaling relation is given in Appendix A.3. Applying Proposition 1 with reference objective h_η and then Proposition 2 yields

$$r_{\mathcal{A}}(t) \le \underbrace{\frac{\Delta_\xi(c - \kappa h_\eta)}{Q}}_{\text{reference mismatch}} + \underbrace{\frac{\kappa G_t(h_p)}{Q}}_{\text{optimization residual}} + \underbrace{\frac{\kappa \lambda L_g n}{Q} \sqrt{\frac{\mathcal{E}_{\mathrm{BCE}}(p; \eta)}{2}}}_{\text{membership error}}, \tag{12}$$

for every $\kappa > 0$. This connects membership supervision to true regret while separating what it does not control. Conditional excess BCE bounds the last term, but accurate memberships alone do not eliminate reference-objective mismatch or incomplete optimization. Lower empirical training BCE therefore does not by itself imply lower test AURC.

Better decisions at later deadlines can still be offset by delayed decision availability. Suppose a guided pipeline returns a fallback with regret r_{fb} until its first new decision is available at τ, and then has regret $r_g(t)$. Let $r_0(t)$ be a baseline profile, and let A_g and A_0 denote their continuous AURCs. With $\tau_c = \min\{t_{\max}, \max\{t_{\min}, \tau\}\}$,

$$A_g < A_0 \quad \Longleftrightarrow \quad \int_{\tau_c}^{t_{\max}} [r_0(t) - r_g(t)]\, \frac{\mathrm{d}t}{t} > \int_{t_{\min}}^{\tau_c} [r_{\mathrm{fb}} - r_0(t)]\, \frac{\mathrm{d}t}{t}. \tag{13}$$

The left integral is the net benefit after a guided decision becomes available; the right is the early net cost of returning the fallback instead of the baseline decision. Guidance improves continuous AURC exactly when the former exceeds the latter. The checkpoint estimate need not recover the switching time τ. Proofs and derivations are given in Appendix A.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Tasks and evaluation cohorts. We evaluate three contextual benchmarks and a separate retrospective energy-scheduling case. Synthetic MDKP has 120 items and 10 resource constraints. Contextualized OR-Library MDKP uses 250 items and 5 constraints and retains public constraint structures with an added contextual objective process (Chu & Beasley, 1998). Its 100 test episodes have held-out contextual seeds but reuse all 30 public base structures. Contextualized Set Cover has 1,500 binary variables and 750 covering constraints in the main experiment. Its structures come from an Ecole-compatible NumPy port of the `SetCoverGenerator` (Prouvost et al., 2021; Balas & Ho, 1980). Each main benchmark uses 600 training, 200 development, and 100 test episodes with three model seeds. The energy case uses price observations from the Irish Single Electricity Market Operator through PredOpt (Ifrim et al., 2012; Mandi et al., 2024). It has five trained models and 32 evaluation dates and is distinct from the larger Energy ablation cohort. Dataset construction and cohort definitions appear in Appendix D.

Baselines and computation budgets. We compare against coefficient-only PTO, SPO+ (Elmachtoub & Grigas, 2022; Mandi et al., 2020), perturbation-gradient DFL (PG) (Gupta & Huang, 2024), learning-to-rank DFL (LTR) (Mandi et al., 2022), stabilized LANCER (Zharmagambetov et al., 2023), and ConPaS (Huang et al., 2024). These comparisons concern the evaluated task-specific implementations. All numerical comparisons use Gurobi 11.0.3 with one solver thread. Node limits are $\{1, 8, 64, 512\}$. The strict E2E grids in Table 1 charge feature processing, inference, guidance, feasible initialization, model construction, search, extraction, and verification to the same deadline. Each model and configuration remain fixed across budgets. Late candidates are not credited, and online selection never uses the true objective coefficients.

Metrics and aggregation. We report normalized regret and log-budget AURC in percent, with lower values preferred. The continuous criterion and checkpoint estimator are defined in Eqs. (3) and (4). Within each seed, episode, method, and budget cell, we first take the median regret across three timing repetitions. We then compute AURC for each reduced seed–episode profile and average regrets and AURCs over seeds and episodes. Figure bands use a dependence-aware cluster bootstrap that preserves method pairing and the task-specific dependence structure described in Appendix D.5. The main table reports point estimates rather than significance claims. One OR-Library episode uses a certified feasible reference with a reported optimality gap of 0.011954%. Its score is reference-based rather than exact hindsight regret. Reference details appear in Appendix D.

5.2 DECISION QUALITY UNDER COMPUTATION BUDGETS

Benchmark results. Table 1 shows that SPP has the lowest reported node and E2E AURCs on OR-Library and Set Cover. On Synthetic MDKP, its E2E AURC is 6.04%, compared with 6.18% for PTO and 5.99% for LTR. SPP has the lowest reported regret at the largest time budget, whereas LTR has the lower integrated regret. On OR-Library, SPP reduces E2E AURC from PTO's 6.24% to 6.13%. These differences refer to the evaluated implementations and their reported point estimates.

The largest reduction relative to PTO occurs on Set Cover. Node AURC decreases from 14.63% to 11.64%, and E2E AURC decreases from 13.93% to 11.24%. The latter is a reduction of 2.69 percentage points, or approximately 19.3% relative to PTO. SPP also has lower regret at each measured E2E budget, from 11.53% at 2 seconds to 11.01% at 16 seconds, compared with PTO's 14.46% and 13.36%.

Effect of early decision availability. The retrospective Energy case illustrates the distinction between late-budget quality and integrated performance. SPP returns the fallback at 0.3 seconds, with regret 6.80%, and achieves 2.39% at the remaining deadlines. Its E2E AURC is therefore 3.10%, compared with 3.08% for PTO and 2.64% for SPO+. The later improvement does not compensate for the early penalty on this budget grid.

Budget profiles. Figure 1 shows the decisions at the measured checkpoints. MDKP profiles change little over the tested intervals, whereas Set Cover retains a lower-regret SPP profile. These

Table 1: **Budgeted decision quality across contextual optimization tasks.** Mean true regret and normalized log-budget AURC (%; lower is better). Bold and underline indicate the best and second-best point estimates, not statistical significance. T_1–T_4 are, in dataset order, $0.03, 0.1, 0.3, 1$; $0.1, 0.3, 1, 3$; $2, 4, 8, 16$; and $0.3, 0.9, 3, 9$ seconds. $> 10^3$ denotes a finite value above 1000%, not a timeout.

		Node budget					Strict E2E time budget				
Dataset	Method	$R@1$	$R@8$	$R@64$	$R@512$	AURC	$R@T_1$	$R@T_2$	$R@T_3$	$R@T_4$	AURC
Synthetic MDKP	PTO	6.16	6.17	6.16	6.13	6.16	6.26	6.23	6.13	6.09	6.18
	SPO+	7.20	7.20	7.21	7.23	7.21	7.24	7.21	7.21	7.22	7.22
	PG	7.29	7.29	7.28	7.28	7.28	7.29	7.27	7.28	7.28	7.28
	LTR	**5.94**	**5.95**	**5.95**	**5.97**	**5.95**	**6.10**	**5.96**	**5.95**	5.99	**5.99**
	LANCER	17.45	17.45	17.44	17.44	17.44	17.49	17.45	17.44	17.44	17.45
	ConPaS	6.36	6.36	6.34	6.31	6.35	6.43	6.40	6.30	6.30	6.36
	Ours	6.04	6.05	6.05	6.04	6.05	6.13	6.07	6.00	**5.97**	6.04
Contextualized OR-Library MDKP	PTO	6.23	6.23	6.23	6.22	6.23	6.28	6.22	6.24	6.23	6.24
	SPO+	7.65	7.66	7.66	7.64	7.65	7.67	7.65	7.64	7.64	7.65
	PG	8.00	8.00	8.00	7.97	7.99	8.02	7.98	7.93	7.92	7.96
	LTR	6.29	6.29	6.27	6.28	6.28	6.32	6.29	6.27	6.26	6.28
	LANCER	22.67	22.68	22.67	22.65	22.67	22.61	22.62	22.59	22.58	22.60
	ConPaS	6.63	6.63	6.62	6.64	6.63	6.64	6.60	6.58	6.57	6.60
	Ours	**6.19**	**6.19**	**6.16**	**6.16**	**6.17**	**6.16**	**6.17**	**6.10**	**6.10**	**6.13**
Contextualized Set Cover	PTO	14.86	14.86	14.52	14.14	14.63	14.46	14.20	13.69	13.36	13.93
	SPO+	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
	PG	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
	LTR	32.86	32.87	32.71	32.21	32.71	33.50	33.08	31.85	31.16	32.42
	LANCER	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
	ConPaS	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
	Ours	**11.73**	**11.73**	**11.68**	**11.31**	**11.64**	**11.53**	**11.32**	**11.12**	**11.01**	**11.24**
Energy scheduling	PTO	3.08	3.08	3.08	3.08	3.08	3.08	3.08	3.08	3.08	3.08
	SPO+	2.65	2.65	2.64	2.64	2.64	**2.64**	2.64	2.64	2.64	**2.64**
	PG	6.48	6.48	6.48	6.48	6.48	6.48	6.48	6.48	6.48	6.48
	LTR	5.38	5.36	5.36	5.36	5.36	5.37	5.36	5.36	5.36	5.36
	LANCER	14.59	14.59	14.60	14.56	14.58	14.57	14.56	14.56	14.56	14.56
	ConPaS	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
	Ours	**2.39**	**2.39**	**2.39**	**2.39**	**2.39**	6.80	**2.39**	**2.39**	**2.39**	3.10

are comparisons of delivered decision quality. They do not measure runtime to a common target quality or the quality of unobserved intermediate decisions.

5.3 Ablation Studies

Objective shaping and initialization. Table 2 compares the four deployment cells in Eq. (E.1) at a shared checkpoint. Objective shaping accounts for most of the observed reduction relative to coefficient-only deployment. On Set Cover, shaping alone yields an AURC of 11.212%, close to the full pipeline's 11.200%, whereas initialization alone yields 16.322%. On OR-Library, shaping alone gives 6.127%, compared with 6.134% for the full pipeline. The main empirical contribution is therefore the objective preference, with no consistent additional reduction from the primal-guided initializer.

Information in the scores. Within-instance shuffling raises AURC from 6.134% to 14.248% on OR-Library and from 11.200% to 17.352% on Set Cover. The correspondence between scores and variables matters in these comparisons. Constant scores also have higher AURC on the first three tasks, but match learned scores on Energy at the displayed precision. Constant scores can still induce a cardinality preference, as explained in Appendix A.4.

These are within-ablation comparisons, not comparisons between the independently configured main and ablation runs. Appendix E reports all score and contextual-input controls, including

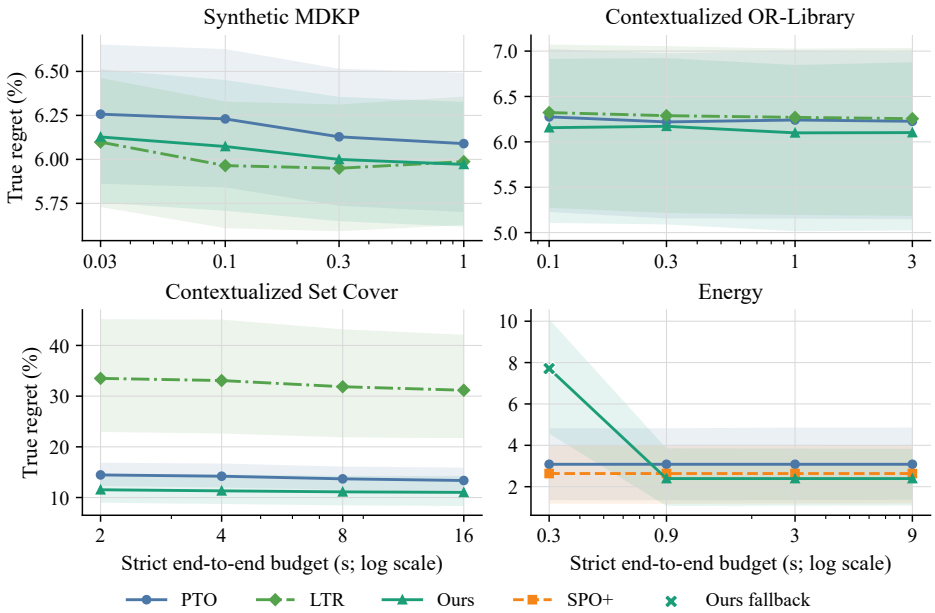


Figure 1: **True-regret profiles under strict end-to-end budgets.** Points are mean regrets after the median reduction of three timing repetitions within each evaluation cell. Bands are 95% dependence-aware cluster-bootstrap intervals. The first three panels use the 3×100 main cohorts. Energy uses the separate retrospective 5×32 cohort. The cross marks SPP's registered fallback at 0.3 seconds. Lines connect measured budgets without smoothing or hindsight selection. The figure displays selected methods from Table 1.

Table 2: **Deployment and score ablations.** Strict E2E AURC (%), with lower values preferred. Each task uses its own 3×100 ablation cohort. Deployment controls share a checkpoint within each seed.

Configuration	Synthetic	OR-Library	Set Cover	Energy
Deployment controls				
B_0 Coefficient-only deployment	6.151	6.431	16.379	11.571
B_1 Objective shaping only	6.044	6.127	11.212	11.542
B_2 Primal initialization only	6.150	6.434	16.322	11.571
B_3 SPP	6.047	6.134	11.200	11.542
Score controls relative to learned scores in B_3				
Within-instance shuffle	6.480	14.248	17.352	12.938
Constant scores	6.374	7.984	12.067	11.542

evaluator-only oracle diagnostics and coefficient-fidelity measurements. Their results do not support a uniform benefit from every contextual input or from joint training.

6 Conclusion

We studied contextual optimization through the decisions available under end-to-end computation budgets. SPP transfers cached solution information into objective preferences without restricting the feasible set or modifying the solver. The analysis separates objective alignment from incomplete optimization and identifies the cost of delayed decision availability. Experiments show the strongest gains on Set Cover, while the retrospective energy case demonstrates that early fallback costs can offset better decisions at later deadlines. Deployment ablations identify objective shaping as the main source of the observed gains. The supervised loss does not directly optimize AURC, and the value of guidance remains dependent on label quality, problem structure, and online overhead.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable Optimization as a Layer in Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145. PMLR, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

Egon Balas and Andrew Ho. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: A computational study. In *Mathematical Programming Studies*, volume 12, pp. 37–60. 1980. doi: 10.1007/BFb0120886.

Dishank Bansal, Ricky T. Q. Chen, Mustafa Mukadam, and Brandon Amos. TaskMet: Task-driven Metric Learning for Model Learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91a5742235f70ae846436d9780e9f1d4-Abstract-Conference.html.

Omar Bennouna, Jiawei Zhang, Saurabh Amin, and Asuman E. Ozdaglar. Contextual Optimization Under Model Misspecification: A Tractable and Generalizable Approach. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3749–3775. PMLR, 2025. URL https://proceedings.mlr.press/v267/bennouna25a.html.

Senne Berden, Ali İrfan Mahmutoğulları, Dimos Tsouros, and Tias Guns. Solver-Free Decision-Focused Learning for Linear Optimization Problems. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e7dd11514cc84c83622e4f5766f9eafa-Abstract-Conference.html.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with Differentiable Perturbed Optimizers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9508–9519, 2020. URL https://papers.nips.cc/paper/2020/file/6bb56208f672af0dd65451f869fedfd9-Paper.pdf.

Timo Berthold. Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6): 611–614, 2013. doi: 10.1016/j.orl.2013.08.007. URL https://www.sciencedirect.com/science/article/pii/S0167637713001181.

Antonia Chmiela, Elias Khalil, Ambros Gleixner, Andrea Lodi, and Sebastian Pokutta. Learning to Schedule Heuristics in Branch and Bound. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cb7c403aa312160380010ee3dd4bfc53-Abstract.html.

P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4(1):63–86, 1998. doi: 10.1023/A:1009642405419. URL https://link.springer.com/article/10.1023/A:1009642405419.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based End-to-end Model Learning in Stochastic Optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3fc2c60b5782f641f76bcefc39fb2392-Abstract.html.

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

Aaron M. Ferber, Taoan Huang, Daochen Zha, Martin Schubert, Benoit Steiner, Bistra Dilkina, and Yuandong Tian. SurCo: Learning Linear SURrogates for COmbinatorial Nonlinear Optimization Problems. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10034–10052. PMLR, 2023. URL https://proceedings.mlr.press/v202/ferber23a.html.

Maxime Gasse, Didier Chetelat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact Combinatorial Optimization with Graph Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d14c2267d848abeb81fd590f371d39bd-Abstract.html.

Vishal Gupta and Michael Huang. Decision-Focused Learning with Directional Gradients. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/907a9fb75a408f6c3a2ae1bf84c39e44-Abstract-Conference.html.

Taoan Huang, Aaron M. Ferber, Yuandong Tian, Bistra Dilkina, and Benoit Steiner. Searching Large Neighborhoods for Integer Linear Programs with Contrastive Learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13869–13890. PMLR, 2023. URL https://proceedings.mlr.press/v202/huang23g.html.

Taoan Huang, Aaron M. Ferber, Arman Zharmagambetov, Yuandong Tian, and Bistra Dilkina. Contrastive Predict-and-Search for Mixed Integer Linear Programs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19757–19771. PMLR, 2024. URL https://proceedings.mlr.press/v235/huang24f.html.

Georgiana Ifrim, Barry O'Sullivan, and Helmut Simonis. Properties of energy-price forecasts for scheduling. In *Principles and Practice of Constraint Programming (CP 2012)*, volume 7514 of *Lecture Notes in Computer Science*, pp. 957–972. Springer, 2012. doi: 10.1007/978-3-642-33558-7_68.

Elias B. Khalil, Christopher Morris, and Andrea Lodi. MIP-GNN: A Data-Driven Framework for Guiding Combinatorial Solvers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10219–10227, 2022. doi: 10.1609/aaai.v36i9.21262. URL https://ojs.aaai.org/index.php/AAAI/article/view/21262.

Haoyang Liu, Jie Wang, Zijie Geng, Xijun Li, Yuxuan Zong, Fangzhou Zhu, Jianye Hao, and Feng Wu. Apollo-MILP: An Alternating Prediction-Correction Neural Solving Framework for Mixed-Integer Linear Programming. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ebcdb4d0372ebbdd0e6a7dce9662b6b4-Abstract-Conference.html.

Jayanta Mandi, Emir Demirović, Peter J. Stuckey, and Tias Guns. Smart Predict-and-Optimize for Hard Combinatorial Optimization Problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1603–1610, 2020. doi: 10.1609/aaai.v34i02.5521. URL https://ojs.aaai.org/index.php/AAAI/article/view/5521.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-Focused Learning: Through the Lens of Learning to Rank. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14935–14947. PMLR, 2022. URL https://proceedings.mlr.press/v162/mandi22a.html.

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

Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. Contrastive Losses and Solution Caching for Predict-and-Optimize. In *Proceedings*

of the Thirtieth International Joint Conference on Artificial Intelligence, pp. 2833–2840. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/390. URL https://www.ijcai.org/proceedings/2021/390.

Max B. Paulus and Andreas Krause. Learning to Dive in Branch and Bound. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6bbda0824bcc20749f21510fd8b28de5-Abstract-Conference.html.

Antoine Prouvost, Justin Dumouchelle, Maxime Gasse, Didier Chételat, and Andrea Lodi. Ecole: A library for learning inside milp solvers, 2021.

Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. Decision-Focused Learning without Decision-Making: Learning Locally Optimized Decision Losses. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0904c7edde20d7134a77fc7f9cd86ea2-Abstract-Conference.html.

Chunlin Sun, Shang Liu, and Xiaocheng Li. Maximum Optimality Margin: A Unified Approach for Contextual Linear Programming and Inverse Linear Programming. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32886–32912. PMLR, 2023. URL https://proceedings.mlr.press/v202/sun23e.html.

Yunhao Tang, Shipra Agrawal, and Yuri Faenza. Reinforcement Learning for Integer Programming: Learning to Cut. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9367–9376. PMLR, 2020. URL https://proceedings.mlr.press/v119/tang20a.html.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of Blackbox Combinatorial Solvers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

Jiaqi Yang, Enming Liang, Zicheng Su, Zhichao Zou, Peng Zhen, Jiecheng Guo, Wanjing Ma, and Kun An. DFF: Decision-Focused Fine-Tuning for Smarter Predict-Then-Optimize with Limited Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26868–26876, 2025. doi: 10.1609/aaai.v39i25.34891. URL https://ojs.aaai.org/index.php/AAAI/article/view/34891.

Arman Zharmagambetov, Brandon Amos, Aaron Ferber, Taoan Huang, Bistra Dilkina, and Yuandong Tian. Landscape Surrogate: Learning Decision Losses for Mathematical Optimization Under Partial Information. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/574f145eac328cc4aaf9358e27120eb5-Abstract-Conference.html.

Shlomo Zilberstein. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*, 17(3):73–83, 1996. doi: 10.1609/aimag.v17i3.1232. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1232.

A Proofs and Additional Theoretical Results

We prove Propositions 1 and 2 and derive the soft-prior and latency results. Fix a nonempty feasible set $\mathcal{Z} = \mathcal{Z}(\xi) \subseteq \{0,1\}^n$ and suppress instance arguments. Write $v^\star(a) = \min_{z\in\mathcal{Z}} a^\top z$, $Q = |v^\star(c)| + \varepsilon > 0$, and $L = \log(t_{\max}/t_{\min}) > 0$. All logarithms are natural. Throughout, z_t denotes the feasible decision actually available at t, including the fallback. The residual $G_t(a) = a^\top z_t - v^\star(a)$ always refers to this same decision, even when a differs from the deployed objective.

A.1 Proof of Proposition 1

Let $z_c^\star \in \arg\min_{z\in\mathcal{Z}} c^\top z$. For every $\kappa > 0$,

$$\begin{aligned} c^\top(z_t - z_c^\star) &= (c-\kappa d)^\top(z_t - z_c^\star) + \kappa d^\top(z_t - z_c^\star) \\ &\le \Delta_\xi(c - \kappa d) + \kappa G_t(d). \end{aligned} \tag{A.1}$$

The first term is bounded by the feasible-set width because both z_t and $z_c^\star$ are feasible. The second term is at most $\kappa G_t(d)$ because $d^\top z_c^\star \ge v^\star(d)$. Division by Q proves Eq. (9).

For fixed d and κ across deadlines, the mismatch term does not depend on t. Integration with respect to the normalized measure $\mathrm{d}t/(Lt)$ therefore gives

$$\mathrm{AURC}_{\mathcal{A}} \le \frac{\Delta_\xi(c-\kappa d) + \kappa\overline{G}(d)}{Q}, \qquad \overline{G}(d) = \frac{1}{L}\int_{t_{\min}}^{t_{\max}} G_t(d)\,\frac{\mathrm{d}t}{t}. \tag{A.2}$$

The same bound holds for the checkpoint estimator in Eq. (4), with $\overline{G}(d)$ replaced by the corresponding trapezoidal average of $G_t(d)$. Its quadrature weights are nonnegative and sum to one. Neither result requires monotonicity of true regret or optimality of the returned decision.

A.2 Proof of Proposition 2

Fix (x,ξ) and let $\eta_j = \mathbb{E}[Y_j \mid x,\xi]$ under the label-generation procedure. Take $p \in (0,1)^n$. If $\mathcal{R}_{\mathrm{BCE}}(p)$ denotes the conditional expected mean binary cross-entropy, then

$$\mathcal{R}_{\mathrm{BCE}}(p) - \mathcal{R}_{\mathrm{BCE}}(\eta) = \frac{1}{n}\sum_{j=1}^{n} \mathrm{KL}\big(\mathrm{Bern}(\eta_j)\,\|\,\mathrm{Bern}(p_j)\big) = \mathcal{E}_{\mathrm{BCE}}(p;\eta). \tag{A.3}$$

This identity follows by taking the expectation of each binary loss. It does not require independence among the labels. Boundary values of η_j are interpreted by continuity, with $0\log 0 = 0$.

For fixed p_j, the Bernoulli KL as a function of its first argument a has second derivative $1/[a(1-a)] \ge 4$ on $(0,1)$, and its value and first derivative vanish at $a = p_j$. Hence $\mathrm{KL}(\mathrm{Bern}(\eta_j)\|\,\mathrm{Bern}(p_j)) \ge 2(\eta_j - p_j)^2$. Cauchy–Schwarz then yields

$$\|p-\eta\|_1 \le \sqrt{n\sum_{j=1}^{n}(p_j-\eta_j)^2} \le n\sqrt{\mathcal{E}_{\mathrm{BCE}}(p;\eta)/2}. \tag{A.4}$$

Set $\lambda = \gamma/(1-\gamma)$, $h_p = u - \lambda g(p)$, and $h_\eta = u - \lambda g(\eta)$. Let $z_\eta \in \arg\min_{z\in\mathcal{Z}} h_\eta^\top z$. Since $h_\eta = h_p + \lambda[g(p) - g(\eta)]$,

$$\begin{aligned} 0 \le h_\eta^\top z_t - v^\star(h_\eta) &= h_p^\top(z_t - z_\eta) + \lambda[g(p)-g(\eta)]^\top(z_t - z_\eta) \\ &\le G_t(h_p) + \lambda\|g(p) - g(\eta)\|_1 \\ &\le G_t(h_p) + \lambda L_g\|p-\eta\|_1 \\ &\le G_t(h_p) + \lambda L_g n\sqrt{\mathcal{E}_{\mathrm{BCE}}(p;\eta)/2}. \end{aligned} \tag{A.5}$$

The first upper bound uses $h_p^\top z_\eta \ge v^\star(h_p)$ and $\|z_t - z_\eta\|_\infty \le 1$. The next uses the assumed ℓ_1-Lipschitz property of the complete vector map g, and the last follows from Eq. (A.4). This proves Proposition 2 without a coordinatewise assumption on g.

A.3 Connection to True Decision Regret

Proposition 2 bounds suboptimality under h_η, not directly under c. Applying Proposition 1 with reference objective h_η and then using Eq. (A.5) gives, for every $\kappa > 0$,

$$r_{\mathcal{A}}(t) \le \frac{\Delta_\xi(c - \kappa h_\eta) + \kappa G_t(h_\eta)}{Q} \tag{A.6}$$

$$\le \frac{\Delta_\xi(c - \kappa h_\eta) + \kappa G_t(h_p) + \kappa\lambda L_g n\sqrt{\mathcal{E}_{\mathrm{BCE}}(p;\eta)/2}}{Q}. \tag{A.7}$$

This is Eq. (12). The corresponding AURC bound follows by averaging the residual over deadlines. The mismatch and membership-error terms are fixed for the instance. The same argument applies to the normalized checkpoint quadrature.

Relation to the deployed objective. Since $\tilde{d}_\gamma = (1-\gamma)h_p$, for $h_p \neq 0$ we have $d_\gamma = \rho_\gamma h_p$, where $\rho_\gamma = (1-\gamma)/a((1-\gamma)h_p) > 0$. If $h_p = 0$, then $d_\gamma = 0$ and we take $\rho_\gamma = 1$. In either case,

$$v^\star(d_\gamma) = \rho_\gamma v^\star(h_p), \qquad G_t(h_p) = \rho_\gamma^{-1} G_t(d_\gamma). \tag{A.8}$$

Thus the residual in Eq. (A.7) is a rescaled residual of the deployed objective at the same returned z_t. This identity does not assert that separate solver runs under h_p and d_γ follow the same finite-budget trajectory.

Score-transformation conditions. For example, $g(p) = p$ has $L_g = 1$, and $g(p) = 2p - \mathbf{1}$ has $L_g = 2$. The vector assumption also permits mean-centering: for $g(p) = p - \bar{p}\mathbf{1}$, where $\bar{p} = n^{-1}\mathbf{1}^\top p$, the triangle inequality gives $\|g(p) - g(q)\|_1 \le \|p - q\|_1 + |\mathbf{1}^\top(p-q)| \le 2\|p-q\|_1$.

For clipped logits, fix $0 < \delta < 1/2$ and define $\bar{p}_j = \min\{1-\delta, \max\{\delta, p_j\}\}$ and $g_\delta(p)_j = \log[\bar{p}_j/(1-\bar{p}_j)]$. Clipping is nonexpansive, and the derivative of logit on $[\delta, 1-\delta]$ is at most $1/[\delta(1-\delta)]$. Consequently, g_δ is ℓ_1-Lipschitz with $L_g \le 1/[\delta(1-\delta)]$. Any further normalization of the score vector requires a regularity bound for the resulting map. The positive rescaling after objective fusion is instead accounted for by Eq. (A.8).

Conditional excess BCE controls only the membership-estimation term. The label marginals need not define an objective aligned with the realized c, and guidance can change the time available for optimization. The bound therefore does not establish true-regret consistency or a monotonic relation between empirical training loss and test AURC.

A.4 Soft-Prior Representation and Decision Ordering

For finite scores $s = g(p)$, define

$$q_s(z \mid x, \xi) = \frac{\exp(s^\top z)}{Z_s}, \qquad Z_s = \sum_{v \in \mathcal{Z}} \exp(s^\top v). \tag{A.9}$$

Because $\mathcal{Z}$ is finite and nonempty, $0 < Z_s < \infty$ and every feasible decision has positive probability. Moreover,

$$u^\top z - \lambda \log q_s(z \mid x, \xi) = (u - \lambda s)^\top z + \lambda \log Z_s. \tag{A.10}$$

The last term is independent of z. Since $\tilde{d}_\gamma = (1-\gamma)(u - \lambda s)$ and $\mathcal{N}$ preserves exact minimizers, Eq. (7) follows. If $\tilde{d}_\gamma = 0$, both objectives are constant over $\mathcal{Z}$. The partition function is never needed by the algorithm. This representation does not imply that BCE fits q_s or that p equals its constrained marginals.

For feasible alternatives z^+ and z^-, let $\Delta z = z^+ - z^-$ and suppose $c^\top \Delta z < 0$ but $m = u^\top \Delta z > 0$. Then

$$\tilde{d}_\gamma^\top \Delta z = (1-\gamma)m - \gamma s^\top \Delta z = (1-\gamma)(m - \lambda s^\top \Delta z). \tag{A.11}$$

Positive normalization preserves the sign of this difference. The shaped objective therefore strictly prefers z^+ exactly when $\lambda s^\top \Delta z > m$; equality produces a tie. No feasible alternative is removed.

Constant scores $s = a\mathbf{1}$ add a cardinality term $-\lambda a \mathbf{1}^\top z$ to the unscaled objective. This term is constant when all feasible decisions have the same cardinality, but otherwise can change decision preferences. Constant scores are therefore not necessarily equivalent to removing deployment guidance.

A.5 Derivation of the Latency-Compensation Condition

Suppose the guided pipeline has regret r_{fb} until its first new decision is available at τ, and regret $r_g(t)$ thereafter. Let $r_0(t)$ be the baseline profile and set $\tau_c = \min\{t_{\max}, \max\{t_{\min}, \tau\}\}$. Splitting the log-time integral at τ_c gives

$$\begin{aligned} A_g - A_0 = \frac{1}{L}\int_{t_{\min}}^{\tau_c} [r_{\mathrm{fb}} - r_0(t)]\,\frac{\mathrm{d}t}{t} \\ + \frac{1}{L}\int_{\tau_c}^{t_{\max}} [r_g(t) - r_0(t)]\,\frac{\mathrm{d}t}{t}. \end{aligned} \tag{A.12}$$

Since $L > 0$, $A_g < A_0$ exactly when the accumulated benefit after τ_c exceeds the early cost relative to the baseline, which proves Eq. (13). Empty intervals contribute zero.

Constant-regret special case. Suppose $r_0(t) = r_b$ and the guided regret after availability is constant at r_g, with $r_{\mathrm{fb}} > r_b > r_g$. Let $w_\tau = \log(\tau_c/t_{\min})/L \in [0, 1]$. Then $A_0 = r_b$ and $A_g = w_\tau r_{\mathrm{fb}} + (1 - w_\tau) r_g$, so

$$A_g < A_0 \quad \Longleftrightarrow \quad w_\tau < \frac{r_b - r_g}{r_{\mathrm{fb}} - r_g}. \tag{A.13}$$

Thus later decision gains compensate for the initial delay only when the fallback occupies a sufficiently small fraction of the log-budget interval. With random latency or fallback quality, the fallback contribution to expected AURC is $\mathbb{E}[w_\tau r_{\mathrm{fb}}]$. A latency percentile alone does not determine this quantity. These identities concern continuous AURC under the stated availability model. Sparse checkpoint quadrature need not equal the continuous integral, even for this step-shaped profile.

B Optimization Tasks and Benchmark Construction

The four benchmarks comprise two multidimensional knapsack settings, contextualized set cover, and a retrospective energy-scheduling case. All tasks use the minimization convention in Eq. (1). The distinction between observable context, known constraints, and hidden objective coefficients is maintained during training, deployment, and evaluation.

B.1 Synthetic Multidimensional Knapsack

An instance contains n items and m resource constraints. Let q_j denote the realized profit of item j, $W_{rj} \geq 0$ its consumption of resource r, and $b_r \geq 0$ the corresponding capacity. The decision $z_j \in \{0, 1\}$ indicates whether item j is selected. The optimization problem is

$$\begin{aligned} \min_{z \in \{0,1\}^n} \quad & -\sum_{j=1}^{n} q_j z_j \\ \text{subject to} \quad & \sum_{j=1}^{n} W_{rj} z_j \leq b_r, \qquad r = 1, \ldots, m. \end{aligned} \tag{B.1}$$

Thus $c = -q$, $A_\xi = W$, and $b_\xi = b$ in the common formulation. The Synthetic MDKP benchmark uses $n = 120$ and $m = 10$, with 600 training, 200 development, and 100 test episodes. The predictor observes the contextual inputs and known instance information, whereas realized profits are withheld at deployment. Primal labels identify the items selected in the cached training solution. The empty selection is feasible under these packing constraints, although this mathematical fact does not identify the fallback used in a run.

B.2 Contextualized OR-Library Knapsack

The OR-Library benchmark uses the formulation in Eq. (B.1), with $n = 250$ and $m = 5$. It retains public knapsack constraint structures associated with Chu & Beasley (1998) and adds a contextual objective process. The resulting benchmark is a contextualized optimization dataset, not a dataset of originally observed real-world covariates. There are 30 base structures. The 100 test episodes reuse ten held-out structures, so episodes that share a structure are not independent structural samples. Training, development, and test counts are 600, 200, and 100 contextual episodes, respectively.

B.3 Contextualized Set Cover

Let $M \in \{0,1\}^{m \times n}$ be a coverage matrix, where $M_{rj} = 1$ if set j covers element r. The decision $z_j \in \{0,1\}$ selects a set with realized cost c_j. The optimization problem is

$$\begin{aligned} \min_{z \in \{0,1\}^n} \quad & \sum_{j=1}^{n} c_j z_j \\ \text{subject to} \quad & \sum_{j=1}^{n} M_{rj} z_j \geq 1, \qquad r = 1, \ldots, m. \end{aligned} \tag{B.2}$$

This corresponds to $A_\xi = -M$ and $b_\xi = -\mathbf{1}$. The main benchmark has $n = 1{,}500$, $m = 750$, and matrix density 0.03. Structures are generated by the Ecole-compatible NumPy port of `SetCoverGenerator` used in the experiments (Prouvost et al., 2021; Balas & Ho, 1980). There are 600 training, 200 development, and 100 test episodes. Primal labels record the selected sets in a cached feasible cover.

When every row has a nonzero entry, selecting every column gives a feasible cover. An incomplete cover is not a feasible fallback. Moreover, shaped costs may be negative even when the original costs are nonnegative. Initialization and local improvement must therefore check the shaped objective rather than assume that removing a redundant column always improves it.

B.4 Retrospective Energy Scheduling

The energy case uses Irish Single Electricity Market Operator price observations distributed through PredOpt (Ifrim et al., 2012; Mandi et al., 2024). For date ℓ, let $z \in \{0,1\}^{n_\ell}$ denote the scheduling variables in the implemented binary model. Let c_ℓ^{E} be its realized coefficient vector and ξ_ℓ its known scheduling requirements. At the common matrix level, the task is

$$\min_{z \in \{0,1\}^{n_\ell}} (c_\ell^{\mathrm{E}})^\top z \quad \text{subject to} \quad A_{\xi_\ell}^{\mathrm{E}} z \leq b_{\xi_\ell}^{\mathrm{E}}. \tag{B.3}$$

The scheduling model includes assignment, temporal, and resource requirements. Historical prices support training and retrospective evaluation. The realized coefficient vector and reference solution are not available to the deployed policy. Primal labels refer to the binary variables of this same scheduling model.

The retrospective main case uses five trained models and 32 dates. The larger Energy ablation uses three trained models and 100 test dates. Their splits and configurations are reported separately in Appendix D.

C Algorithms and Implementation Details

C.1 Offline Label Construction and Training

Algorithm 1 separates offline decision-label construction from gradient updates. The coefficient targets and cached membership labels are those defined in Section 4.1. The algorithm specifies the method-level workflow. Implementation-dependent preprocessing and selection settings must match the reported configurations.

Label construction may require optimization, but no optimizer is called inside the gradient-update loop. Training labels remain fixed after construction. Test coefficients, test reference solutions, and test regrets are excluded from model selection. The counts in Table 17 describe available labels rather than the total optimization workload of every baseline.

C.2 Deadline-Aware Deployment

Algorithm 2 makes the deadline-availability rule explicit. The worker and deadline-controller blocks describe a logical contract, not evidence that a particular asynchronous implementation was used. A new decision must be fully extracted, checked, and ready for return before it can replace the fallback. The deployed routine selects candidates using the surrogate objective and never the hidden true cost.

Algorithm 1 Offline training of Soft Primal Prior

Require: Training data $\mathcal{D}_{\mathrm{tr}}$, labeled indices $\mathcal{I}_{\mathrm{lab}}$, training schedule, and loss weights α, β
Ensure: Trained coefficient and primal heads and a fixed deployment configuration
for $i \in \mathcal{I}_{\mathrm{lab}}$ **do**
  Obtain or load a cached feasible decision $\widetilde{z}_i$ under (c_i, ξ_i)
  Verify feasibility and store $y_i \leftarrow \widetilde{z}_i$ with label-quality metadata
end for
Construct the prescribed features and normalized coefficient targets $\overline{c}_i$
Initialize the shared network and its two output heads
for each training epoch **do**
  for each training minibatch $\mathcal{B}$ **do**
    Compute coefficient outputs u_i and logits a_i for $i \in \mathcal{B}$
    Evaluate the coefficient and available-label primal losses in Eq. (5)
    Update all trainable parameters with Adam
  end for
end for
Apply the development selection procedure to the model and deployment settings
Fix the selected model, preprocessing, g, $\mathcal{N}$, γ, and initializer
return the fixed pipeline configuration

Algorithm 2 SPP deployment under an end-to-end deadline

Require: Observable (x, ξ), fixed trained pipeline, budget t, and an admissible feasible-fallback procedure
Ensure: A feasible decision available at the requested deadline
Start the external clock at T_0 and set the deadline $D \leftarrow T_0 + t$
Establish the registered fallback z_{fb} with any online cost charged
Initialize the new-decision record as unavailable
Budgeted worker
Construct features and compute u and p
Set $s \leftarrow g(p)$ and $d \leftarrow \mathcal{N}((1-\gamma)u - \gamma s)$
Construct a feasible start $z^0 \leftarrow H_\xi(d)$
Build the original optimization model with objective d and start z^0
Allocate remaining online time to search, extraction, verification, and return
Run the solver and obtain the candidate selected under d
Verify the candidate against the original constraints and prepare it for return
Publish $(\widehat{z}, T_{\mathrm{ready}})$ only after these operations finish
Deadline controller
if a verified new-decision record is available with $T_{\mathrm{ready}} \leq D$ **then**
  return $\widehat{z}$
else
  return z_{fb}
end if

The evaluated budgets assume that the fallback itself is available in time. A late solver output cannot be backdated to the deadline. An internal solver time limit does not alone implement this contract, because solver termination can require additional work. The implementation must define how the controller observes completed decisions without waiting for a late worker.

C.3 Network Architecture and Training Configuration

The model has a shared multilayer-perceptron trunk and separate coefficient and primal-score heads. Its inputs combine structural descriptors, contextual features, and interactions. In the unified Energy configuration, the 24 inputs comprise 12 contextual period aggregates and 12 scheduling-variable descriptors. Tables 3–5 report the available development-selected configurations. The Energy entries describe the unified configuration and are not substituted for the separate retrospective main-case checkpoints.

Table 3: Network architecture and training schedule. Depth and width refer to the shared trunk. The Energy row refers to the unified configuration.

Task	Inputs	Depth	Width	Parameters	Epochs	Batch size
Synthetic MDKP	26	2	128	20,226	19	16
OR-Library MDKP	21	4	64	14,018	19	4
Set Cover	15	2	128	18,818	21	4
Energy	24	3	64	10,050	27	4

Table 4: Development-selected learning and fusion parameters. The optimizer is Adam. The weights β and α multiply coefficient and primal losses, and γ controls deployment fusion.

Task	Learning rate	Dropout	β	α	γ
Synthetic MDKP	1.739×10^{-4}	0.3	4.7621	9.5505	0.03719
OR-Library MDKP	1.494×10^{-3}	0.3	0.11435	0.08458	0.26766
Set Cover	7.215×10^{-4}	0.0	2.6922	1.4009	0.07146
Energy	4.728×10^{-5}	0.1	4.4936	0.02559	0.49091

C.4 Labels and Numerical Conventions

For logit a and binary target y, the binary cross-entropy can be evaluated stably as $\mathrm{softplus}(a) - ya$. With a shared trunk, both losses update shared parameters. A small objective gap does not imply a small membership error, because several different binary vectors may attain the same objective value. Cached labels therefore identify feasible selections rather than unique required assignments.

The coefficient target transformation and the deployment normalization serve different roles and must be specified separately. Under squared loss, the population target is the conditional mean of the transformed coefficient vector. For a nonlinear transformation $\mathcal{C}$, this is $\mathbb{E}[\mathcal{C}(c) \mid x, \xi]$, which need not equal $\mathcal{C}(\mathbb{E}[c \mid x, \xi])$. The presence of a coefficient head alone does not establish calibration in absolute monetary units. The analysis of coordinatewise score transformations is in Appendix A.2, with the clipped-logit example in Appendix A.3.

C.5 Feasible Initialization

The constructor H_ξ supplies a feasible starting decision using only known constraints and the deployed objective. For packing constraints, the empty selection is feasible, and an insertion preserves feasibility when every resource constraint is checked before acceptance. For set cover, a constructor must cover every row before its output is admissible. Deleting a redundant column preserves feasibility but can increase the shaped objective if that column has negative cost.

A local move from z to z' is objective-nonincreasing only when

$$z' \in \mathcal{Z}(\xi), \qquad d^\top (z' - z) \le 0. \tag{C.1}$$

The Energy constructor must satisfy all scheduling constraints. Neither the zero vector nor the all-one vector is a generic scheduling fallback. These observations establish feasibility conditions, not the implementation-specific ranking rules of H_ξ.

C.6 Deadline Availability and Node-Limited Evaluation

The online clock includes feature processing, inference, score transformation, initialization, model construction, search, extraction, and output verification. The same trained model and deployment settings serve all tested deadlines. A candidate is credited only when ready under Algorithm 2. This empirical availability criterion is not a hard real-time operating-system guarantee.

Node-limited runs use the same learned outputs but bound solver search rather than elapsed online time. Prediction and construction work do not consume solver nodes. Equal node limits therefore do not imply equal runtime or equal total computational work.

Table 5: Deployment transformations and local initialization refinements. "None" refers to the local refinement, not the absence of a feasible start.

Task	Objective normalization	Primal-score transformation	Local refinement
Synthetic MDKP	ℓ_∞	Centered score	One-add
OR-Library MDKP	ℓ_2	Clipped logit	None
Set Cover	ℓ_2	Clipped logit	One-swap
Energy	ℓ_∞	Bounded probability	One-swap

Table 6: Problem sizes, data splits, and evaluation cohorts. Split counts are training/development/test episodes or dates. "Unresolved" identifies conflicting metadata, and "not specified" identifies a missing value. Neither denotes non-applicability.

Task and cohort	Variables	Constraints	Split	Models $\times$ test
Synthetic main	120	10	600/200/100	3×100
Synthetic ablation	120	10	600/200/100	3×100
OR-Library main	250	5	600/200/100	3×100
OR-Library ablation	250	5	600/200/100	3×100
Set Cover main	1,500	750	600/200/100	3×100
Set Cover ablation	1,500	Unresolved	600/200/100	3×100
Energy ablation	Not specified	–	589/100/100	3×100
Energy retrospective	Not specified	–	Train/dev not specified	5×32

D Experimental Protocols

D.1 Data Splits and Evaluation Cohorts

Table 6 separates problem dimensions, data splits, and evaluation cohorts. A model–episode pair is an evaluation unit, not an independently trained model. The first three main benchmarks use three model seeds and 100 test episodes. The retrospective Energy case uses five models and 32 dates. All ablation tables use their own three-model, 100-episode cohorts.

The chronological Energy ablation split has 589 training dates from 2011-11-01 to 2013-06-14, 100 development dates from 2013-06-15 to 2013-09-22, and 100 test dates from 2013-09-23 to 2013-12-31. The retrospective main case instead evaluates 2013-07-27 through 2013-08-27. These dates overlap the development period of the other protocol, so the two experiments must not be presented as a common holdout or pooled into one Energy estimate.

D.2 Baseline Implementations

The comparison includes coefficient-only PTO, SPO+ (Elmachtoub & Grigas, 2022; Mandi et al., 2020), perturbation-gradient DFL (Gupta & Huang, 2024), learning-to-rank DFL (Mandi et al., 2022), LANCER (Zharmagambetov et al., 2023), and ConPaS (Huang et al., 2024). The numerical results concern their task-specific implementations. A method's name alone does not specify its training budget, label requirements, inference overhead, or adaptation to an unknown objective.

D.3 Solver and Execution Environment

The reported solver is Gurobi 11.0.3 with one solver thread. The relative target gap is zero, and the feasibility and integrality tolerances are both 10^{-6}. Solver seeds are fixed by configuration. Setting the solver's internal time limit does not replace the external availability rule. Some runs used parallel execution on a shared host. This can affect wall-clock variability even with one solver thread per run. Repetition medians do not remove every system-level effect.

Table 7: Computation budgets. Times are end-to-end seconds. The node grid is $(1, 8, 64, 512)$ for every task.

Task	t_1	t_2	t_3	t_4
Synthetic MDKP	0.03	0.1	0.3	1
OR-Library MDKP	0.1	0.3	1	3
Set Cover	2	4	8	16
Energy	0.3	0.9	3	9

Table 8: Baseline-specific implementation information needed to interpret the numerical comparison. These fields describe the required run specification, not additional completed experiments.

Method	Configuration fields to report
PTO	Architecture, coefficient target, model-selection criterion, and feasible initializer.
SPO+	Forward and training optimization problems, exact or bounded labels, oracle limits, and completed update count.
PG	Perturbation estimator, direction distribution, step size, oracle limits, and completed update count.
LTR	Candidate-solution pool, ranking loss, pool-construction cost, and candidate selection at deployment.
LANCER	Loss-model architecture, target generation, stabilization, alternation schedule, and completed update count.
ConPaS	Deployment objective source, graph inputs, contrastive targets, search or restriction rule, repair, and fallback.

D.4 Reference Objectives

The theoretical regret uses an exact hindsight optimum. One OR-Library episode instead uses a certified feasible reference with a reported optimality gap of 0.011954%.[1] For reference value v^{ref}, the corresponding score is

$$\widehat{r}_{\mathrm{ref}}(z) = \frac{c^\top z - v^{\mathrm{ref}}}{|v^{\mathrm{ref}}| + \varepsilon}. \tag{D.1}$$

It is reference-based rather than exact hindsight regret. If $v^{\mathrm{lb}} \le v^\star(c) \le v^{\mathrm{ref}}$, then

$$c^\top z - v^{\mathrm{ref}} \le c^\top z - v^\star(c) \le c^\top z - v^{\mathrm{lb}}. \tag{D.2}$$

These are bounds on the unnormalized regret. They do not remove the change in denominator in Eq. (D.1). A score relative to a feasible reference can be negative if a returned solution improves on that reference. Any clipping convention must be reported explicitly rather than inferred from the nonnegativity of exact regret.

D.5 Aggregation and Statistical Analysis

Let $R^{(j)}_{sikm}$ be the percentage regret for model seed s, episode i, budget t_k, method m, and timing repetition j. The reported time-profile center first reduces repetitions and then averages evaluation units,

$$\widetilde{R}_{sikm} = \mathrm{median}_{j=1,2,3}\, R^{(j)}_{sikm}, \qquad \overline{R}_m(t_k) = \frac{1}{SI}\sum_{s=1}^{S}\sum_{i=1}^{I} \widetilde{R}_{sikm}. \tag{D.3}$$

This is a median of regret observations, not selection of the run with median elapsed time. With $L = \log(t_K/t_1)$, define the trapezoidal weights

$$w_1 = \frac{\log(t_2/t_1)}{2L}, \qquad w_K = \frac{\log(t_K/t_{K-1})}{2L}, \qquad w_k = \frac{\log(t_{k+1}/t_{k-1})}{2L} \quad (1 < k < K). \tag{D.4}$$

They are nonnegative and sum to one. The dataset AURC in percent is

$$\widehat{A}_m = \frac{1}{SI}\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{k=1}^{K} w_k \widetilde{R}_{sikm}. \tag{D.5}$$

[1] The episode identifier is `orlib-11-context-840342`.

Table 9: Solver settings supported by the current experimental description.

Setting	Value
Solver version	Gurobi 11.0.3
Solver threads	1
Relative target gap	0
Feasibility tolerance	10^{-6}
Integrality tolerance	10^{-6}
Timing repetitions	3 per time-evaluation cell
Node and E2E limits	Table 7

The same log-budget weights apply to the node grid after replacing t_k by the node limit. Arithmetic averaging commutes with this quadrature, but taking medians across episodes does not generally commute with it. Actual deadline outputs are retained without hindsight replacement by a decision with lower true cost at another budget.

The figure intervals use 10,000 crossed cluster-bootstrap replicates with method pairing preserved. Model seeds are resampled together with instances for Synthetic and Set Cover, base structures for OR-Library, and seven-date blocks for Energy. Timing repetitions are reduced first and are not counted as independent test observations. The dependence-aware OR-Library and Energy intervals replace the earlier seed–episode resampling for these figures. With only three or five model seeds, the intervals describe variation supported by those models. For a paired comparison of methods m and m', the effect of interest is

$$\widehat{D}_{m,m'} = \frac{1}{SI} \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{k=1}^{K} w_k (\widetilde{R}_{sikm} - \widetilde{R}_{sikm'}). \tag{D.6}$$

A negative difference favors m. A paired interval must preserve the same resampling indices for both methods. The result tables below report point estimates and do not assert statistical significance from the presence or absence of overlap between marginal figure intervals.

D.6 Additional Budget Profiles

E Extended Ablation Studies

E.1 Deployment Controls and Effect Definitions

Within each model seed, the four deployment controls share a reconstructed checkpoint with coefficient output u and primal score s. Let $d = \mathcal{N}((1-\gamma)u - \gamma s)$. Each pair below specifies the solver objective and feasible start,

$$\begin{aligned} B_0 &= (\mathcal{N}(u), H_\xi(\mathcal{N}(u))), \quad & B_1 &= (d, H_\xi(\mathcal{N}(u))), \\ B_2 &= (\mathcal{N}(u), H_\xi(d)), & B_3 &= (d, H_\xi(d)). \end{aligned} \tag{E.1}$$

The network, fusion parameters, constructor, and solver settings remain fixed apart from these stated deployment choices. All work actually executed is included in E2E time, even if a control ignores an output. B_0 is not the separately trained PTO model in the main comparison.

Let A_j be the AURC of B_j. The baseline-conditional improvements are $\mathsf{A}_0 - \mathsf{A}_1$ for objective shaping and $\mathsf{A}_0 - \mathsf{A}_2$ for initialization. The averaged factorial effects are instead

$$\begin{aligned} E_{\text{obj}} &= \tfrac{1}{2}[(\mathsf{A}_0 - \mathsf{A}_1) + (\mathsf{A}_2 - \mathsf{A}_3)], \\ E_{\text{init}} &= \tfrac{1}{2}[(\mathsf{A}_0 - \mathsf{A}_2) + (\mathsf{A}_1 - \mathsf{A}_3)]. \end{aligned} \tag{E.2}$$

The interaction uses the cost convention $I_{\text{cost}} = \mathsf{A}_0 - \mathsf{A}_1 - \mathsf{A}_2 + \mathsf{A}_3$. A positive value means that the joint improvement is smaller than the sum of the baseline-conditional improvements. It does not mean positive synergy. The same definitions apply to pointwise regret differences.

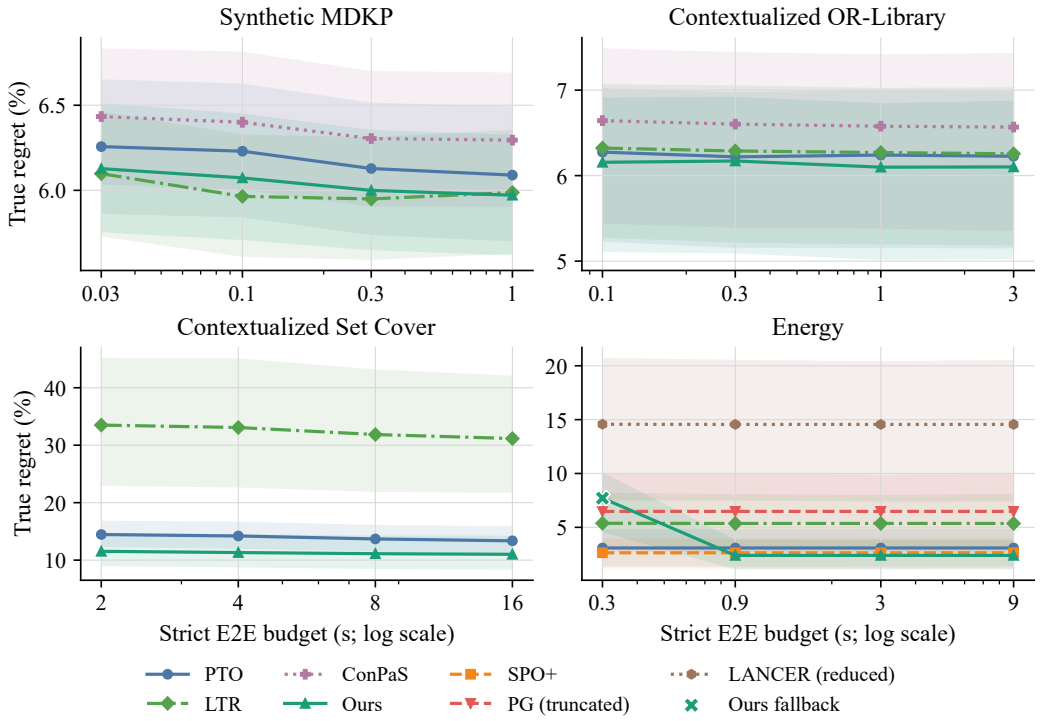


Figure 2: Additional strict E2E regret profiles. The first three panels use the main 3×100 cohorts, and Energy uses the retrospective 5×32 cohort. Bands are 95% dependence-aware cluster-bootstrap intervals. Lines connect measured budgets. The cross marks an available fallback. Only selected configurations are plotted. Reduced or truncated variants must retain the same identifiers as their numerical table rows, and failed direct outputs are distinguished from valid fallback outputs.

Table 10: Strict E2E AURC (%) for deployment and primal-score controls. Lower is better. All columns use separate 3×100 ablation cohorts. Oracle scores are non-deployable diagnostics.

Control	Synthetic	OR-Library	Set Cover	Energy
Deployment controls				
B_0 Coefficient-only deployment	6.151	6.431	16.379	11.571
B_1 Objective shaping only	6.044	6.127	11.212	11.542
B_2 Primal initialization only	6.150	6.434	16.322	11.571
B_3 SPP	6.047	6.134	11.200	11.542
Primal-score controls				
C_1 Within-instance shuffle	6.480	14.248	17.352	12.938
C_2 Cross-instance transfer	6.444	13.679	17.384	11.577
C_3 Constant scores	6.374	7.984	12.067	11.542
C_4 Oracle scores	3.033	0.403	0.459	0.000

E.2 Deployment and Primal-Score Results

Tables 10 and 11 separate E2E and node AURCs for readability. Every column uses its own 3×100 ablation cohort, including Energy. Learned-score control C_0 and full-context control D_0 both reproduce B_3, so their identical rows are not repeated. The score controls are within-instance shuffling (C_1), cross-instance transfer (C_2), constant scores (C_3), and evaluator-only oracle scores (C_4).

Objective shaping explains most of the deployment improvement in the reported point estimates. Initialization alone has little effect, and its additional benefit on top of objective shaping is not consistent across tasks. Learned scores improve on the shuffled and constant controls for the first

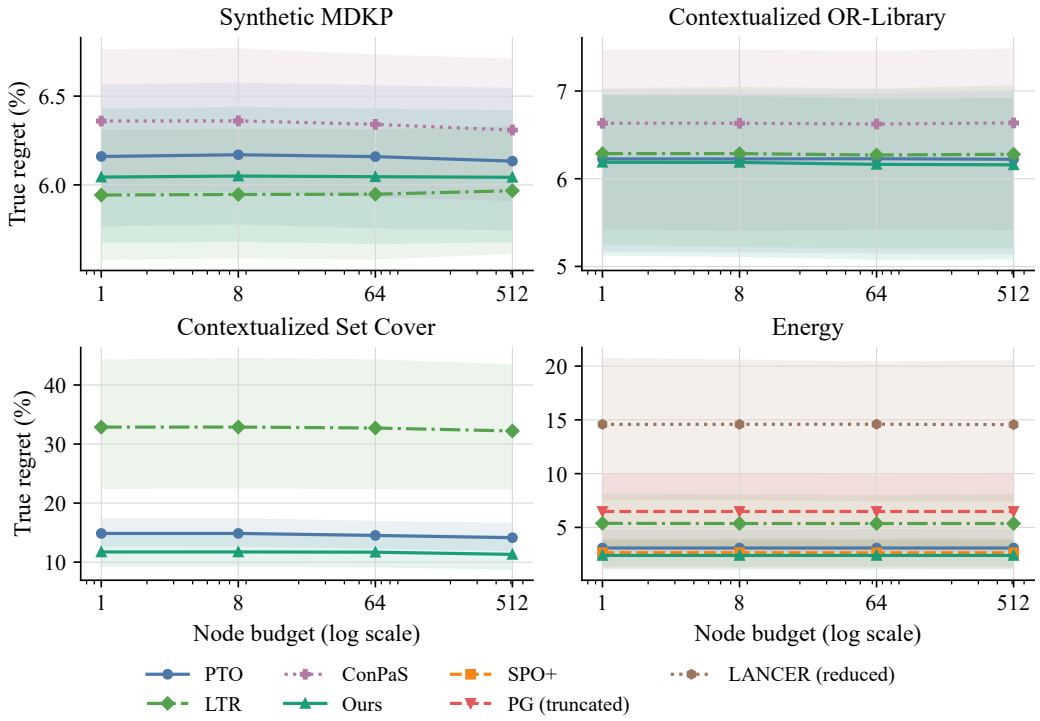


Figure 3: Additional node-budget regret profiles on $(1, 8, 64, 512)$. The cohorts and plotted configurations correspond to those in Fig. 2. Node limits bound search rather than end-to-end time and therefore do not charge prediction or construction latency. Lines connect measured budgets without a monotonicity correction.

Table 11: Node AURC (%) for the controls in Table 10. The node grid is $(1, 8, 64, 512)$. Lower is better.

Control	Synthetic	OR-Library	Set Cover	Energy
Deployment controls				
B_0 Coefficient-only deployment	6.162	6.432	16.960	11.543
B_1 Objective shaping only	6.032	6.185	11.807	11.542
B_2 Primal initialization only	6.162	6.432	16.850	11.543
B_3 SPP	6.027	6.187	11.711	11.542
Primal-score controls				
C_1 Within-instance shuffle	6.497	14.290	18.204	12.938
C_2 Cross-instance transfer	6.439	13.725	18.171	11.577
C_3 Constant scores	6.339	7.978	12.683	11.542
C_4 Oracle scores	3.034	0.405	0.459	0.000

three tasks, but match constant scores on Energy at the displayed precision. A constant score can still create a cardinality preference, as shown in Appendix A.4. Oracle scores use information unavailable at deployment and are not a general upper bound on achievable performance.

E.3 Contextual-Input Controls

Table 12 reports the available contextual-input controls. D_0 is the full model, D_1 omits context, D_2 shuffles context, D_3 uses base profits for OR-Library, and D_4 and D_5 restrict the relevant branch to structure or context, respectively. These are task-dependent information tests rather than a uniform

Table 12: Contextual-input ablations. AURC is in percent, with lower values preferred. A dash denotes a non-applicable control, not a failed run. D_0 is identical to B_3.

Control	Synthetic	OR-Library	Set Cover	Energy
Panel A. Strict E2E AURC				
D_0 Full context	6.047	6.134	11.200	11.542
D_1 No context	5.935	6.414	14.690	11.344
D_2 Shuffled context	6.122	6.430	14.164	11.982
D_3 Base-profit only	–	6.411	–	–
D_4 Structure-only branch	6.046	6.410	11.279	11.535
D_5 Context-only branch	6.378	7.307	13.907	10.592
Panel B. Node AURC				
D_0 Full context	6.027	6.187	11.711	11.542
D_1 No context	5.926	6.455	14.753	11.344
D_2 Shuffled context	6.110	6.475	14.280	11.977
D_3 Base-profit only	–	6.455	–	–
D_4 Structure-only branch	6.051	6.481	11.675	11.535
D_5 Context-only branch	6.370	7.332	14.440	10.592

Table 13: Coefficient fidelity under matched training controls. MAE uses each task's coefficient units and is not comparable across tasks. The entries are point estimates, not equivalence or non-inferiority tests.

Task	Predictor	nMSE	MAE	Spearman ρ
Synthetic	Coefficient-only A_0	0.02961	0.17435	0.87422
	Dual-head coefficient A_1	0.02993	0.17519	0.87188
OR-Library	Coefficient-only A_0	0.12136	0.33774	0.50745
	Dual-head coefficient A_1	0.12398	0.34662	0.49674
Set Cover	Coefficient-only A_0	0.00814	0.08114	0.98971
	Dual-head coefficient A_1	0.00697	0.07464	0.99146
Energy	Coefficient-only A_0	0.99990	97,563.63	0.56686
	Dual-head coefficient A_1	0.99990	97,563.63	0.56686

ranking of input modalities. Some restricted-input models improve on the full model on Synthetic or Energy. The results therefore do not establish that every contextual input improves decision quality.

E.4 Coefficient Fidelity

The training controls compare a separately trained coefficient-only model A_0 with the coefficient head of the dual-head model A_1. They use the matched ablation configuration rather than the independently tuned PTO settings. For example, the matched coefficient-only control has E2E AURCs of 6.47 on OR-Library and 20.22 on Set Cover, whereas the main PTO results are 6.24 and 13.93. This cross-configuration difference cannot be attributed to primal supervision.

Coefficient fidelity does not improve uniformly across tasks. Similar point estimates do not establish formal non-inferiority. The Energy MAE is retained in its reported units rather than rescaled to match the magnitudes of other tasks.

E.5 Scope of the Ablations

The ablations were designed after inspection of the main results and fixed before their own evaluation. Deployment controls share a reconstructed checkpoint within each model seed, but this checkpoint is not asserted to be bitwise identical to an unpersisted main-run checkpoint. The results support within-ablation comparisons rather than numerical attribution across different configurations and cohorts. No completed test-set sensitivity study for γ is reported. Development search trials do not constitute such a study.

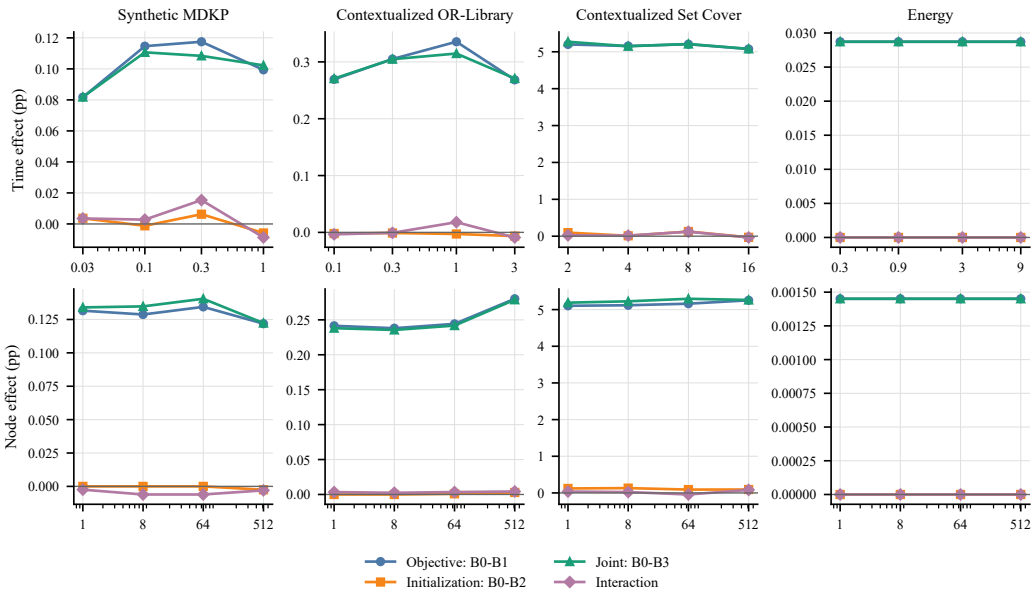


Figure 4: Pointwise deployment effects in percentage points. The objective, initialization, and joint curves are $R_0 - R_1$, $R_0 - R_2$, and $R_0 - R_3$, respectively. The interaction is $R_0 - R_1 - R_2 + R_3$, so a positive value denotes subadditivity of the improvements. These are baseline-conditional pointwise differences, not AURC factorial effects. Energy uses the separate 3×100 ablation cohort.

Table 14: Extended-budget endpoints on the 3×100 ablation cohorts. The node limit is 2,048. The E2E limits are 6 seconds for OR-Library and 32 seconds for Set Cover. Regret and solver gap are percentages. The solver gap refers to the objective supplied to that solver.

		Extended node budget		Extended E2E budget	
Task	Control	Regret	Solver gap	Regret	Solver gap
OR-Library	B_0	6.443	0.155	6.353	0.080
	B_1	6.133	0.127	6.099	0.003
	B_2	6.447	0.155	6.383	0.081
	B_3	6.134	0.128	6.098	0.003
Set Cover	B_0	16.146	1.701	15.971	0.941
	B_1	10.989	0.734	10.926	0.297
	B_2	16.106	1.710	15.990	0.902
	B_3	10.937	0.770	10.934	0.288

F Extended-Budget Results and Computational Costs

F.1 Extended-Budget Endpoints

Table 14 gives the available extended-budget endpoints for OR-Library and Set Cover. Each task has one additional node limit and one additional E2E deadline. These measurements are endpoint regrets and solver gaps, not AURCs or complete long-budget profiles. The shaped-objective controls retain lower true regret at these endpoints, whereas initialization-only controls remain close to coefficient-only deployment. Nonzero solver gaps prevent an exact-convergence claim. Even a zero gap for a shaped objective would not certify optimality under the hidden true objective.

Table 15: Online time components in seconds. P95 is the 95th percentile of preprocessing time after repetition reduction. All other time columns report medians. The Energy cohort is the retrospective 5×32 case.

Task	Method	Preprocess	P95	Setup	Search	Extract	E2E
Synthetic	PTO	0.0059	0.0084	0.0013	0.1377	0.0003	0.1451
	ConPaS	0.0085	0.0107	0.0016	0.1055	0.0003	0.1173
	SPP	0.0044	0.0047	0.0013	0.0957	0.0003	0.1021
OR-Library	PTO	0.0144	0.0249	0.0018	0.6008	0.0004	0.6230
	ConPaS	0.0277	0.0394	0.0022	0.2764	0.0004	0.3024
	SPP	0.0105	0.0106	0.0018	0.2883	0.0004	0.3016
Set Cover	PTO	0.4695	0.5554	0.0059	3.5267	0.0011	4.0035
	LTR	0.4513	0.5691	0.0059	3.5060	0.0011	4.0029
	SPP	0.5174	0.6204	0.0058	0.5718	0.0011	1.0996
Energy	PTO	0.1463	0.1533	0.0040	0.0543	0.0007	0.2074
	SPO+	0.1460	0.1551	0.0040	0.0525	0.0007	0.2055
	SPP	0.3303	0.3444	0.0039	0.0531	0.0007	0.3857
	ConPaS	0.1525	0.2050	0.0042	0.0041	0.0002	0.1616

Table 16: Availability outcomes in the retrospective Energy case. Fallback counts are divided by the number of evaluated cells at the listed deadlines. A direct-output failure can coexist with a valid fallback decision.

Method	E2E deadlines (s)	Cells	Fallbacks	Reason
SPP	0.3	160	160	Guided decision unavailable at the deadline
SPP	0.9, 3, 9	480	0	Guided decision available
ConPaS	0.3, 0.9, 3, 9	640	640	Direct-output failure

F.2 Online Time Components

Table 15 reports the available component summaries. Except for the preprocessing 95th percentile, values are medians after the documented repetition reduction and aggregation across evaluated budgets. Medians of components need not sum to the median total. These measurements are not runtimes to a common target regret and do not establish a matched-quality speedup. For retrospective Energy, SPP's raw preprocessing P95 is 0.352 seconds, whereas its P95 after repetition reduction is 0.3444 seconds. Neither percentile identifies which individual decisions were available at a particular deadline. Availability is determined per evaluation cell.

F.3 Decision Availability and Failure Status

The original event pairs are separated from timing statistics because fallback use and direct-output failure have different meanings. Table 16 reports the Energy events with explicit denominators. Each unit is a model–date–budget cell after repetition reduction, not a separate training run.

For SPP, all 160 model–date pairs use the fallback at the first deadline, and later deadlines admit the guided solution. Its fallback frequency is therefore 100% at 0.3 seconds and 25% over the four equally counted budget cells. ConPaS has direct-output failure at all four deadlines. The reported zero-event entries for the first three tasks and for PTO and SPO+ in Energy indicate no recorded fallback or direct-output-failure events for those listed configurations.

A valid fallback is part of the deployed pipeline and must remain in its decision-quality evaluation. It must not be relabeled as a successful direct output. A cell without any valid deadline-available decision is not a finite-regret observation. Likewise, a large finite regret is not a timeout. The displayed symbol $> 10^3$ is appropriate only for an actually computed finite percentage above that threshold.

Table 17: Training-label coverage. "Exact" denotes a decision reported as certified optimal under the true training objective. Bounded decisions are feasible incumbents from limited solves. Counts are labels, not total oracle calls or elapsed training time. NR means not reported, not zero.

Task	Episodes	Exact	Bounded	Detail
Synthetic MDKP	600	600	0	All training decisions exact
OR-Library MDKP	600	599	1	One bounded incumbent
Set Cover	600	215	385	Bounded solves use 512 nodes
Energy	589	NR	NR	Label-quality breakdown not reported

F.4 Offline Label Construction and Training Cost

After label construction, SPP performs no optimizer calls during gradient updates. This does not eliminate offline label costs. Complete measured training times and solver-call totals are not available for every method and configuration, so no cross-method training-speed or amortization claim is made from these counts.

Projected workloads. The configuration-based workload estimates exceed 115,000 oracle or target solves for the full SPO+ and LANCER schedules on the 600-episode tasks and 113,000 on the chronological Energy split. The corresponding PG estimates exceed 230,000 and 226,000 calls. The Set Cover ConPaS workload estimate includes at least 5.18 billion sparse-edge traversals. These are estimates for particular local schedules, not measured completed-run costs or intrinsic requirements of the published methods. Reduced and two-epoch variants must remain explicitly identified.