POOLDFL: SOLVER-FREE DECISION-FOCUSED LEARNING WITH AUGMENTED SOLUTION POOLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-focused learning often requires solving an optimisation problem at every training step. We introduce POOLDFL, which augments a shared pool of training optima with feasible neighbours and keeps the pool fixed throughout training. Neighbour enumeration requires no additional solver calls, and each update evaluates a restricted SPO+ loss through matrix products and maximisation over the pool. This loss recovers the full objective whenever the pool contains a global optimum of its inner optimisation problem. Experiments on shortest path, assignment and knapsack show that augmentation improves decision quality over pools containing only training optima. On 10×10 shortest path, POOLDFL has a mean regret gap of -0.02% relative to SPO+ over 35 paired seeds. On assignment, it satisfies a 1% non-inferiority criterion after Holm correction. Gradient diagnostics associate closer alignment with the full objective with smaller regret gaps.

1 INTRODUCTION

In predict-then-optimise problems, a model predicts a cost vector and an optimisation oracle selects the downstream decision. Prediction errors matter through the decisions they induce, motivating decision-focused learning (DFL) (Donti et al., 2017; Wilder et al., 2019). The convex SPO+ surrogate (Elmachtoub & Grigas, 2022) directly targets decision quality, but evaluates an optimisation oracle at every gradient step. Solver-free alternatives use precomputed solution geometry or direct prediction losses to avoid these repeated solves (Berden et al., 2025; Tang & Khalil, 2024b; Lee et al., 2026; Wan & Liu, 2026).

A loss evaluated over a finite candidate set compares the label only with the alternatives in that set. If a relevant decision is missing, changing the scoring rule does not make it available to the update. A shared feasible set offers another source of candidates: the optimal label for one training instance is also feasible for every other instance. These labels can therefore be reused across the training set.

Figure 1 illustrates why this reuse matters on a 3×3 grid. Scoring the true path and its unit-square neighbour gives zero restricted SPO+ loss, although the predicted path has positive regret. A label from another training instance supplies the missing probe optimum, recovering the full loss and a nonzero update. The example motivates a pool that draws candidates from the training set as a whole.

POOLDFL builds this pool from training optima and expands it with their feasible one-hop neighbours. Enumeration adds candidates without further solver calls. The resulting pool extends fixed-pool solution caching (Mulamba et al., 2021) and remains fixed during training. Each update scores the candidates using the SPO+ objective and selects a maximiser through a matrix product and a row-wise maximum.

We establish conditions under which the restricted objective recovers the full SPO+ loss and a matching subgradient. Experiments on three tasks compare the base and augmented pools, with controlled comparisons of candidate composition and scoring. We also measure agreement between pool-based and full-objective subgradients during training; across nine configurations, closer alignment is associated with smaller regret gaps.

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.

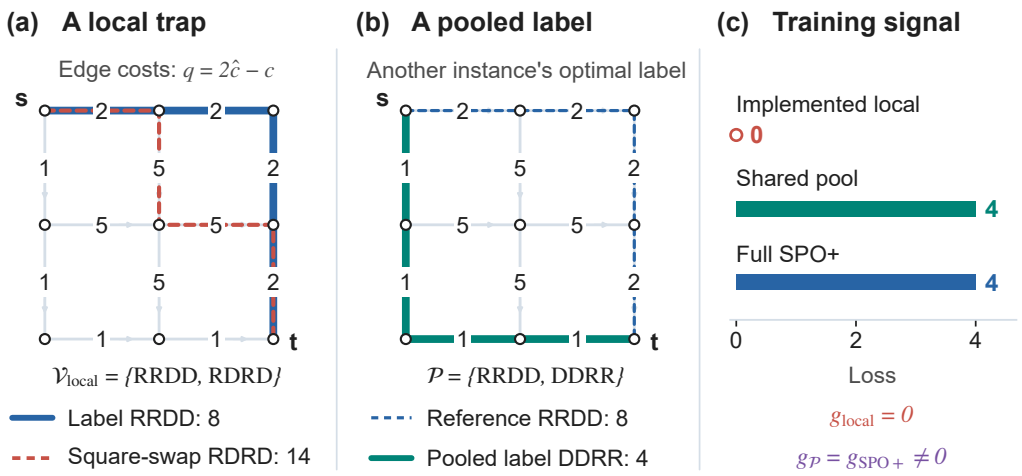

Figure 1: **A missing candidate can eliminate the training signal.** A constructed 3×3 right-down shortest-path example. Both grids show the same probe costs $q = 2\hat{c} - c$. (a) The label RRDD has probe cost 8; its only implemented unit-square-swap neighbour RDRD has cost 14, so the restricted loss and gradient are zero. (b) A pooled label DDRR recovers the unique global probe optimum, of cost 4. (c) Pooled and full SPO+ losses are both 4, with identical nonzero subgradients. The predicted decision's unnormalised regret is 2. The unit-square neighbourhood is a subset of full LP adjacency. The true costs and all six feasible paths are given in Appendix B.

2 Related Work

Surveys by Mandi et al. (2024); Sadana et al. (2025) review the main approaches to DFL.

Solution reuse. Mulamba et al. (2021) initialise a cache with training optima for contrastive learning and SPO+ caching; Mandi et al. (2022) develop ranking losses over candidate solutions. The base PoolDFL pool uses SPO+ caching with solve probability $p_{\text{solve}} = 0$; its augmented pool adds enumerated neighbours of the cached optima before training begins. Other approaches reduce solve costs using relaxations and warm starts (Mandi et al., 2020) or dual information (Rodriguez-Diaz et al., 2025).

Optimality geometry and solver-free losses. Maximum optimality margin learns from observed solutions through optimality conditions (Sun et al., 2023). LAVA uses adjacent vertices (Berden et al., 2025); CAVE aligns predictions with the optimality cone (Tang & Khalil, 2024b); PEAR projects errors onto a tangent space (Lee et al., 2026); and WISE uses norm-weighted regression (Wan & Liu, 2026). PoolDFL evaluates a restricted SPO+ objective over training labels and their feasible neighbours, shared across instances with the same feasible set.

Optimisation layers. Differentiable optimisation includes convex and interior-point layers (Amos & Kolter, 2017; Agrawal et al., 2019; Mandi & Guns, 2020), integer layers (Ferber et al., 2020; Paulus et al., 2021), and blackbox, perturbed or implicit-likelihood constructions (Vlastelica et al., 2020; Berthet et al., 2020; Niepert et al., 2021). PoolDFL computes a surrogate subgradient by selecting a cached feasible solution, without differentiating through a solver.

Surrogate design and statistical theory. Prior work establishes generalisation bounds for decision regret (El Balghiti et al., 2019), calibration and excess-risk bounds for SPO+ (Liu & Grigas, 2021), and prediction-loss conditions for downstream risk (Ho-Nguyen & Kılınç-Karzan, 2022). Fenchel–Young losses provide convex surrogate constructions (Blondel et al., 2020); learned local losses and surrogate training of optimisation layers adapt objectives to decision tasks (Shah et al., 2022; Mandi et al., 2025). Our analysis characterises when a finite pool preserves the full SPO+ loss and a matching subgradient.

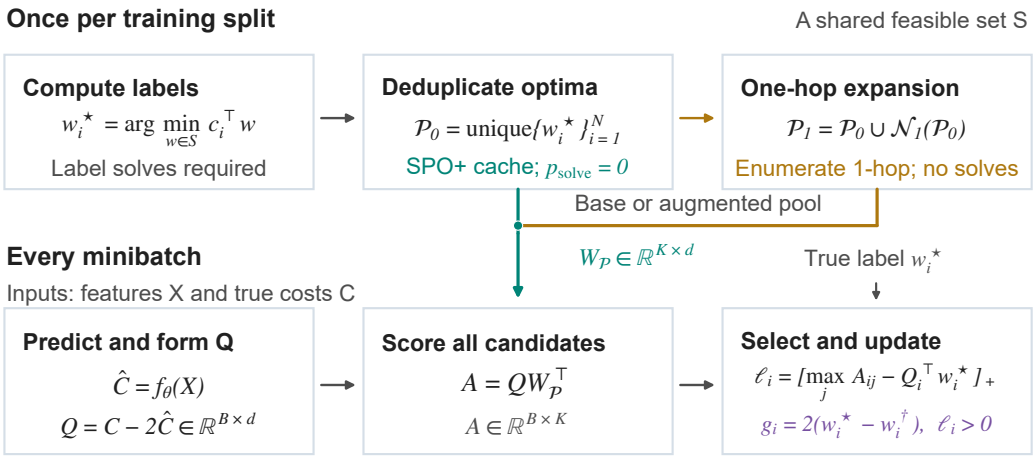

Figure 2: **A fixed candidate pool replaces the training-time oracle.** POOLDFL initialises a pool from training optima and expands it by implemented one-hop enumeration without additional solves. For batch size B, cost dimension d and K candidates, $Q = C - 2\hat{C} \in \mathbb{R}^{B \times d}$ and $W_\mathcal{P} \in \mathbb{R}^{K \times d}$ give scores $A = QW_\mathcal{P}^\top$. The selected candidate determines the positive-loss subgradient; zero is a valid choice at zero loss. Proposition 1 gives the conditions for matching SPO+ updates.

3 SETUP

Let $S \subset \mathbb{R}^n$ be a bounded polytope with vertex set $\mathcal{V}(S)$ and let $w^\star(c) = \arg\min_{w \in S} c^\top w$, with $w^\star = w^\star(c)$ when the cost is clear. Given features x and a predictor $\hat{c} = f_\theta(x)$, the decision regret is

$$R(\hat{c}, c) = c^\top w^\star(\hat{c}) - c^\top w^\star(c) \ \geq \ 0. \tag{1}$$

SPO+ (Elmachtoub & Grigas, 2022) upper-bounds (1) by

$$\ell_{\text{SPO+}}(\hat{c}, c) = \max_{w \in S} \left\{ (c - 2\hat{c})^\top w \right\} + 2\hat{c}^\top w^\star - c^\top w^\star, \tag{2}$$

with subgradient $2(w^\star - w^\star(2\hat{c} - c))$. Solver-free surrogates replace the maximisation oracle in (2).

Regret is normalised by optimal cost throughout, following PyEPO (Tang & Khalil, 2024a).

4 POOLDFL: TRAINING WITH AUGMENTED SOLUTION POOLS

POOLDFL evaluates the SPO+ objective over a pool of cached training solutions and their enumerated neighbours. The pool is constructed once per training split and remains fixed during training. Figure 2 shows its construction and use in each update.

Base solution pool. Let $\mathcal{P}_0 = \{w^\star(c_i)\}_{i=1}^N$, deduplicated over the training split. These optima are the labels used by SPO+ in (2) and by LAVA, PEAR and CAVE. Using $\mathcal{P}_0$ alone gives fixed-pool SPO+ caching with $p_{\text{solve}} = 0$ (Mulamba et al., 2021).

Neighbourhood augmentation. For the implemented one-hop neighbourhood $\mathcal{N}_1(w)$ of a feasible solution w, define

$$\mathcal{P}_1 = \mathcal{P}_0 \ \cup \bigcup_{w \in \mathcal{P}_0} \mathcal{N}_1(w).$$

Enumeration adds feasible candidates without additional solver calls. For grid shortest path, we scan consecutive triples of each cached path and exchange the two sides of a unit square. For assignment, we exchange the jobs of two workers, enumerating $\binom{8}{2} = 28$ transpositions per cached solution. For knapsack, we enumerate feasible item moves and swaps. The resulting solutions are deduplicated with the training optima before scoring. Appendix C details these neighbourhoods.

In the augmentation diagnostics, the pool grows from 977 to 7917 candidates on 10×10 shortest path, from 982 to 19415 on assignment, and from 745 to 13316 on knapsack. We report both $\mathcal{P}_0$ (base) and $\mathcal{P}_1$ (augmented) in the experiments. Test instances contribute no vertices, and held-out regret is measured against a fresh solve.

Restricted training objective. For a candidate pool $\mathcal{P} \subseteq \mathcal{V}(S)$, define

$$\ell_{\mathcal{P}}(\hat{c}, c) = \Big(\max_{w \in \mathcal{P}} (c - 2\hat{c})^\top (w - w^\star) \Big)_+ . \tag{3}$$

The objective applies to either $\mathcal{P}_0$ or $\mathcal{P}_1$. The following proposition relates it to the full loss.

Proposition 1 (Restriction and exact reduction)**.** *For any $\mathcal{P} \subseteq \mathcal{V}(S)$, the loss in* (3) *is convex in $\hat{c}$, $\ell_{\mathcal{P}} \le \ell_{\text{SPO+}}$ pointwise, and $\ell_{\mathcal{P}} = \ell_{\text{SPO+}}$ whenever $\mathcal{P} \ni w^\star(2\hat{c} - c)$. Its subgradient is $2(w^\star - w^\dagger)$ with $w^\dagger$ chosen from the maximisers over $\mathcal{P} \cup \{w^\star\}$. Under the equality condition, choosing the same maximiser as* SPO+ *gives the same subgradient, including compatible choices at zero loss.*

Appendix A proves the result by expressing the hinge as a maximum over $\mathcal{P} \cup \{w^\star\}$. Including the label makes the zero-loss branch part of the same maximisation.

Computation. For batch size B, evaluating (3) requires a $B \times n$ by $n \times |\mathcal{P}|$ matrix product and a row-wise maximum, with $O(Bn|\mathcal{P}|)$ scoring cost and $O(n|\mathcal{P}|)$ storage for the candidate matrix. Label preprocessing, deployment and diagnostic measurements use optimisation oracles; all training updates use the fixed pool.

Exactness check. On a 4×4 grid all 20 vertices can be enumerated. With $\mathcal{P} = \mathcal{V}(S)$, (3) reproduces the SPO+ loss value and matches its subgradient on 100% of samples; summation over the same pool is $2.8\times$ larger.

5 Experiments

5.1 Experimental protocol

Tasks and training. We use shortest path on an $N\times N$ right-down grid, 8×8 assignment, and a 20-item, 2-constraint knapsack. Data follow the Elmachtoub polynomial DGP with degree 4, $\varepsilon = 0.5$ and $n_{\text{train}} = 1000$. Unless stated otherwise, we train a linear predictor for 150 epochs using Adam with learning rate 10^{-3} and batch size 32. Each seed resamples the train/test split and the initialisation; statistics are paired by seed. The tasks have linear objectives and use PyEPO (Tang & Khalil, 2024a).

Baseline configurations. We use LAVA with ϵ=0.1, WISE with its target-normalised, norm-weighted squared loss, and PEAR with normal injection β=0.1, following their published configurations. For CAVE, Moreau's decomposition turns the cone projection into a K-dimensional non-negative least-squares problem requiring no LP solve. The five-seed comparison reports each method's lower mean regret from learning rates $\{10^{-3}, 10^{-2}\}$ (oracle selection). Appendix C gives the formulae and implementation details.

Evaluation. The augmented pool was selected on knapsack and assignment before its 35-seed evaluation on 10×10 shortest path. The non-inferiority tests use a pre-specified margin of 1% of mean SPO+ regret. Appendix I gives the evaluation criteria; Table 9 separates the pilot and confirmatory results.

5.2 Decision quality and training cost

We first compare the augmented pool with SPO+ and the base pool, then place the fixed-pool approach alongside solver-free baselines. Figure 3 shows the quality and runtime comparisons; Appendix G includes the identity and MLP cases.

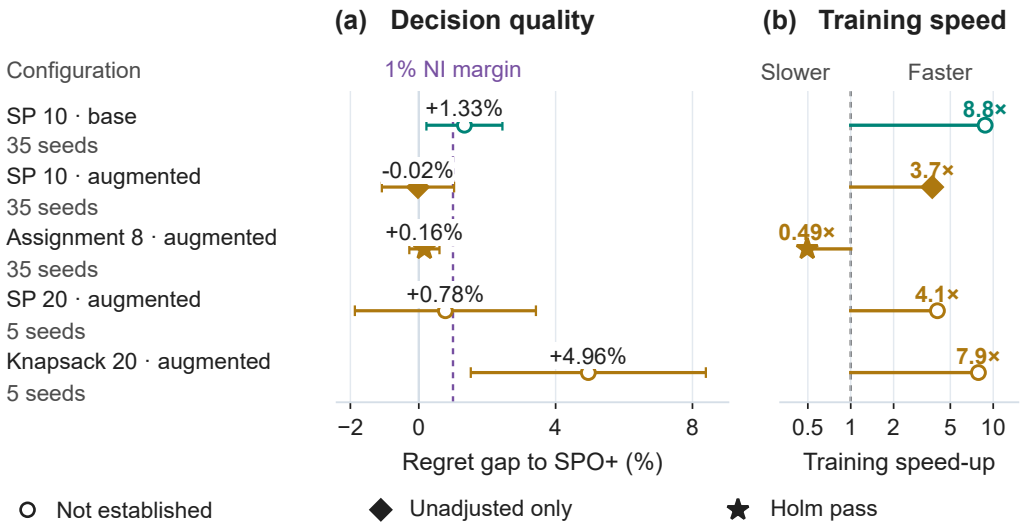

Figure 3: **Decision quality and training speed across candidate pools.** (a) Mean regret gaps and paired two-sided 95% intervals, divided by the corresponding mean SPO+ regret and expressed in percent. The dashed line is the pre-registered 1% non-inferiority margin. Marker shapes indicate non-inferiority outcomes after Holm correction across all nine tests. (b) Ratios of mean training times on a log axis; values above one are faster. Timing ratios are point estimates. Section 5 reports the corresponding unadjusted and corrected tests.

Decision quality with the augmented pool. On 10×10 shortest path over 35 paired seeds, the augmented pool has a mean regret gap of -0.02% relative to SPO+ ($14/35$ wins, $p_{\text{ni}} = 0.0285$). The base pool has a gap of $+1.33\%$ on the same seeds ($6/35$ wins, $p_{\text{ni}} = 0.73$).

Across the full nine-test family, Holm correction (Holm, 1979) retains augmented assignment ($p_{\text{ni}} = 2.5 \times 10^{-4}$) and the 5×5 identity case. Augmented 10×10 shortest path passes the unadjusted test. The two-sided confidence intervals in Figure 3 are distinct from these one-sided tests.

Cross-task augmentation. The augmented pool has lower mean regret than the base pool on all three tasks in Table 1. The gap to SPO+ changes from $+40.9\%$ to $+5.0\%$ on knapsack, from $+1.97\%$ to -0.13% on assignment ($p_{\text{ni}} = 0.005$), and from $+2.61\%$ to $+0.78\%$ on 20×20 shortest path. These comparisons use 5 seeds for knapsack and shortest path and 12 for assignment, separately from the 35-seed confirmations.

Table 1: Effect of one-hop pool augmentation on coverage and regret. Augmentation enumerates neighbours of training optima without additional solver calls.

		Coverage			Base pool		Augmented	
Problem	Seeds	base	aug	SPO+	regret	Δ%	regret	Δ%
Knapsack (20)	5	0.81	0.88	0.0902	0.1271	+40.88%	0.0947	+4.96%
Assignment 8×8	12	0.88	1.00	0.2403	0.2451	+1.97%	0.2400	−0.13%
SP 20×20	5	0.33	0.64	0.1108	0.1137	+2.61%	0.1116	+0.78%

Comparison with solver-free baselines. The base pool also yields lower mean regret than the solver-free baselines in the five-seed 10×10 comparison (Table 2). Each method uses its selected learning rate from $\{10^{-3}, 10^{-2}\}$. PoolDFL has mean regret 0.1262, compared with 0.1235 for SPO+; LAVA, WISE, PEAR and CaVE range from 0.1376 to 0.1822 under their published settings. Appendix C reports parameter sensitivity, and Appendix A.2 derives the two margin-weighting variants included in the comparison.

Table 2: Normalised regret (↓) on 10×10 shortest path, degree 4, $\varepsilon = 0.5$, $n = 1000$, five seeds resampling both split and initialisation. Each method reports its lower mean regret from learning rates $\{10^{-3}, 10^{-2}\}$ (oracle selection). Additional LAVA and PEAR settings assess sensitivity.

Method	Regret	Std	$\eta^\star$	Sec/seed
Regression	0.1343	0.0084	0.01	3.5
LAVA (ϵ=0)	0.1818	0.0144	0.001	9.9
LAVA (ϵ=0.1, published)	0.1822	0.0074	0.001	9.7
WISE	0.1376	0.0086	0.001	2.7
PEAR (β=0)	0.1596	0.0062	0.001	9.6
PEAR (β=0.1, published)	0.1388	0.0064	0.001	16.3
CAVE	0.1798	0.0119	0.001	33.6
RHL (m_j)	0.1443	0.0058	0.001	11.4
RHL (m_j^2, see Appendix A.2)	0.1399	0.0053	0.001	11.7
ARM	0.1459	0.0046	0.001	11.9
SPO+	0.1235	0.0060	0.01	36.0
POOLDFL (base pool)	0.1262	0.0074	0.01	4.9

Training cost. On 10×10 shortest path, the base pool trains 7.3× faster than SPO+ in the tuned comparison of Table 2 and 8.8× faster at the protocol default in Table 9. The augmented pool has a ratio of 3.7×. On 8×8 assignment, the augmented pool's ratio is 0.49×. These times cover the training loop and periodic evaluation, excluding label preprocessing and pool construction. For the 10×10 confirmation, the timing means use 23 seeds for SPO+ and the base pool, and 35 for the augmented pool.

Full-pool identity. On 5×5, the base pool contains all $\binom{8}{4} = 70$ vertices, so the restricted and full SPO+ losses coincide by Proposition 1. The -0.12% regret difference arises from tie-breaking between valid subgradients (Appendix D).

5.3 CANDIDATE-POOL AND SCORING ABLATIONS

Scaling with problem size. With 1000 training labels and $N \in \{5, 10, 15, 20\}$, the base pool's regret gaps are -0.12%, -0.08%, $+2.42\%$ and $+2.76\%$ ($\rho = +1.00$, exact one-sided permutation $p = 0.0417$; Table 10).

Pool size. Subsampling the training optima increases mean regret in the five-seed comparison. Pools of 8–512 vertices have gaps of 11.1–13.7%, compared with 0.02% for the full pool of 970 optima (Table 11, Appendix H).

Candidate composition. At a matched pool size, training optima yield lower mean regret than uniformly sampled feasible solutions in the 12-seed comparison (Tables 12 and 13). The equally large random pool has a regret gap of $+12.6\%$ to SPO+; an 8× larger random pool reduces this gap to $+8.5\%$. Halving the training pool increases the gap to 10.1%. At a matched 500 instances, training optima have a gap of $+10.1\%$ and held-out optima $+15.1\%$; the held-out pool is worse on 10 of 12 seeds; the paired contrast between them is $+0.0067$ with a 95% interval of $[-0.0004, +0.0137]$ and $p = 0.061$.

In (3), a sample's own optimum contributes $(c - 2\hat{c})^\top (w^\star - w^\star) = 0$; masking it leaves the loss and subdifferential unchanged. The training signal comes from the other candidates in the pool (Table 12).

Effect of the scoring rule. Holding the augmented pool fixed, we compare POOLDFL with three forms of contrastive loss (Mulamba et al., 2021). Table 3 reports their mean regrets alongside two-stage regression.

At 12 seeds, summing over the pool instead of selecting gives a 0.9% regret increase with an interval spanning zero. The larger difference from the contrastive variants in Table 3 comes from the score, $c - 2\hat{c}$ rather than $\hat{c}$: the best contrastive variant has mean regret 4.4%, 9.3% and 116% higher than augmented POOLDFL on shortest path, assignment and knapsack, respectively.

Table 3: Comparison of scoring rules on a shared candidate pool. Contrastive solution caching (Mulamba et al., 2021) uses the same pool as augmented PoolDFL, with zero cache growth, in linear, hinged-sum and max forms. Entries are regret with the relative gap to SPO+ in parentheses; lower is better. Shortest path and assignment use 12 seeds; knapsack uses 5. All methods use learning rate 10^{-3}.

Method	SP 10×10	Assign 8×8	Knapsack (20)
Two-stage regression	0.1406 (+5.9%)	0.2745 (+14.2%)	0.1204 (+33.4%)
Contrastive cache, linear (Mulamba et al., 2021)	0.1374 (+3.5%)	0.2623 (+9.1%)	0.2044 (+126.5%)
Contrastive cache, hinged sum	0.1729 (+30.3%)	0.2893 (+20.4%)	0.3135 (+247.4%)
Contrastive cache, max (MAP form)	0.1802 (+35.7%)	0.3294 (+37.1%)	0.2473 (+174.1%)
PoolDFL (augmented pool)	0.1316 (-0.9%)	0.2400 (-0.1%)	0.0947 (+5.0%)

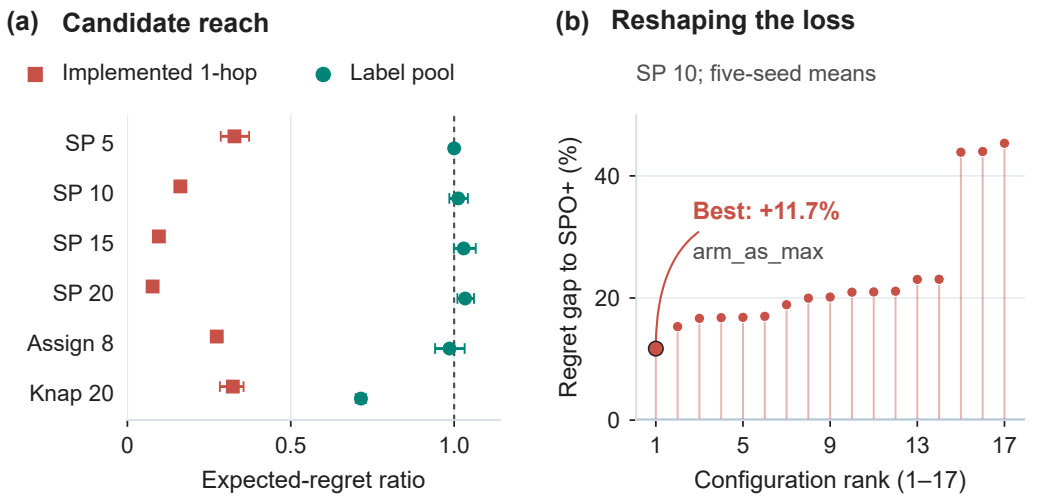

Figure 4: **Implemented-neighbourhood coverage and local-loss regret.** (a) Expected-regret ratios for the implemented one-hop neighbourhood and the base label pool at each problem's measured predictor-residual scale. Error bars are bootstrap 95% intervals over cost instances; the ratio can exceed one. SP N denotes an $N\times N$ grid, Assign 8 an 8×8 assignment, and Knap 20 a 20-item, two-constraint knapsack. (b) All 17 local configurations on 10×10 shortest path, ranked by five-seed mean regret gap to SPO+. The lowest gap is $+11.7\%$. Configuration details and values are in Tables 6 and 8.

5.4 Candidate coverage and local losses

We next examine how candidate restriction affects decisions under perturbed costs and how local losses behave on fixed neighbourhoods. Figure 4 summarises both comparisons.

Coverage and aggregation. For each vertex w_j^{adj} adjacent to $w^\star$ write $v_j = w_j^{\mathrm{adj}} - w^\star$ and $m_j = c^\top v_j > 0$, and let $s_j(\hat{c}) = v_j^\top \hat{c}$. Crossing facet j costs exactly m_j. The adjacent-facet model of expected regret under $\delta \sim \mathcal{N}(0, \sigma^2 I)$ is

$$\mathcal{R}_\Sigma(\sigma) = \sum_j m_j \, \Phi\left(\frac{-m_j}{\sigma \|v_j\|}\right). \tag{4}$$

Equation (4) sums the expected-regret contributions of facet crossings. For each perturbation δ, the restricted optimiser selects one candidate by minimising $(c+\delta)^\top w$. The corresponding expected true-cost regret over a candidate set $\mathcal{V}$ is

$$\mathcal{R}_{\max}(\mathcal{V}) = \mathbb{E}_\delta\left[c^\top\left(\arg\min_{w\in\{w^\star\}\cup\mathcal{V}}(c+\delta)^\top w - w^\star\right)\right], \tag{5}$$

where the expectation uses $\delta \sim \mathcal{N}(0, \sigma^2 I)$. We define *coverage* as the ratio of $\mathcal{R}_{\max}(\mathcal{V})$ to the unrestricted expected regret $\mathbb{E}_\delta[R(c+\delta, c)]$. This diagnostic uses exact solves for its ground truth. Coverage can exceed one because a restricted set may select a vertex with higher true cost than the unrestricted selection; proximity to one measures agreement in expected regret.

The summed model satisfies $\mathcal{R}_\Sigma \geq \mathcal{R}_{\max}$ whenever more than one facet can be crossed at once. Table 7 measures both against Monte-Carlo truth.

Aggregation and coverage errors partly cancel. Equation (4) recovers 0.40–0.52 of true expected regret, while (5) recovers 0.12–0.40. Summation over-counts by 0.11–0.28 of true expected regret on the 1-hop neighbourhood.

The inflation grows with the candidate set. On the 2-hop neighbourhood it is 0.94 to 3.92 of true expected regret, so the summed model over-counts truth by up to $4.1\times$. In the five-seed training comparison of Table 8, widening to 2-hop changes the summed loss's regret gap from $+18.9\%$ to $+16.8\%$, and the selecting loss's gap from $+20.0\%$ to $+21.1\%$.

The discrepancy due to candidate restriction is 2 to 5 times that due to aggregation. Running steepest descent over the 1-hop neighbourhood to local optimality reaches 0.35 to 0.47. Over true LP adjacency local optimality would imply global optimality and the walk would reach 1.0. The measurements here use the implemented subsets of adjacency (Table 5).

Prediction-error scale and true adjacency. The perturbation scale σ in (4) can be set from the data-generating noise or from measured prediction errors. Here the DGP noise is 0.5, while the standard deviation of held-out residuals from a trained linear predictor is 6.8–7.5, roughly $14\times$ larger. The predictor is linear and the true costs are degree-4 polynomials of the features (Appendix E).

The expected-regret ratios depend on both the candidate set and the perturbation scale (Table 6). At the residual scale, the implemented neighbourhood has ratios of 0.08–0.33. The label-pool ratios are 0.99–1.03 on shortest path and assignment, and 0.71 on knapsack.

Table 5 distinguishes implemented neighbourhoods from true LP adjacency: single-cycle differences on the Birkhoff polytope and single-detour differences on the s–t path polytope. Where enumerable, true adjacency recovers 0.85–0.99 of expected regret. The true neighbourhood is $11\times$ larger on 5×5 shortest path and $574\times$ larger on assignment ($16{,}064$ neighbours versus 28 transpositions).

Local-loss comparison. To separate the scoring rule from the candidate set, we vary the loss over fixed implemented neighbourhoods (Appendix F). The adjacent-facet family is

$$L(\hat{c}) = \mathrm{agg}_j \; \omega_j \left[(t_j - s_j)_+\right]^p, \tag{6}$$

with threshold $t_j \in \{0, \epsilon, m_j/2, m_j\}$, aggregation $\mathrm{agg} \in \{\mathrm{sum}, \mathrm{mean}, \max, \text{log-sum-exp}\}$, weight $\omega_j \in \{1, m_j\}$ and $p \in \{1, 2\}$. LAVA is $(\epsilon, \mathrm{sum}, 1, 1)$ and ARM is $(m_j, \mathrm{sum}, m_j, 2)$. The threshold $m_j/2$ follows from the SPO+ probe cost $2\hat{c} - c$, under which $w^\star + v_j$ beats $w^\star$ exactly when $s_j < m_j/2$, so $\max_j (m_j - 2s_j)_+$ is SPO+ restricted to the 1-hop neighbourhood.

The 17 configurations vary the threshold, aggregation, weighting and power, and include 2-hop variants and warm-started pivot walks. Their five-seed mean regret gaps range from $+11.7\%$ to $+45.3\%$ (Table 8). Replacing sum with max in ARM reduces its gap from $+15.3\%$ to $+11.7\%$ on the same neighbourhood, whose coverage is 0.16.

5.5 Gradient fidelity and decision quality

We compare static coverage, the frequency of recovering a global probe optimum, and subgradient cosine against relative regret (Figure 5, Table 4). Static coverage uses $\sigma = 0.5$ here; Figure 4(a) uses the predictor-residual scale.

On 10×10 shortest path, augmentation raises the exact hit rate from 0.029 to 0.157 and gradient cosine from 0.776 to 0.850. On assignment, the corresponding changes are 0.023 to 0.541 and 0.755 to 0.915.

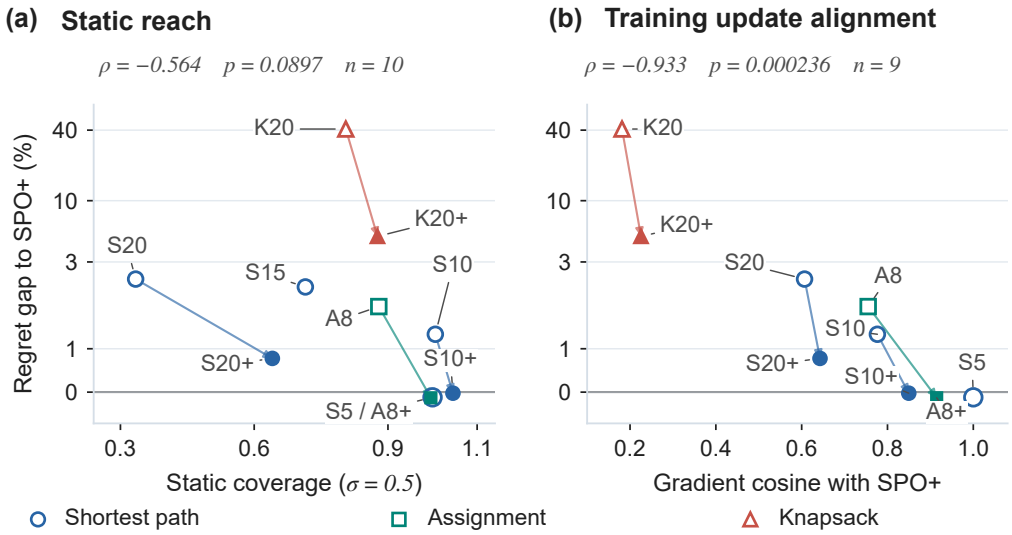

Figure 5: **Static coverage and training-update fidelity differ.** Left: coverage at $\sigma = 0.5$ for ten configurations; right: gradient cosine for the nine with recorded probes. Open/filled markers denote base/augmented pools; arrows connect four matched base–augmented pairs. S, A and K denote shortest path, assignment and knapsack; "+" denotes augmentation. SP 5 is the full-pool identity. SP 15 has coverage measurements only. Assignment augmentation uses the 12-seed diagnostic record; the 35-seed confirmation is reported in Figure 3. Both vertical axes are linear for $|\text{gap}| \leq 3\%$ and logarithmic beyond it. The correlations summarise the association across these heterogeneous configurations. Table 4 reports the diagnostic values.

Across configurations, gradient cosine has the strongest association with regret gaps ($\rho = -0.93$, $p = 0.0002$), followed by exact hit rate ($\rho = -0.84$, $p = 0.005$) and static coverage ($\rho = -0.56$, $p = 0.09$).

The cosine diagnostic uses approximately 10^3 oracle calls, compared with 1.5×10^5 in a full SPO+ training run.

Table 4: Candidate diagnostics and decision quality. Coverage is measured before training; hit rate and gradient cosine are measured on 64 probe instances every 10 epochs. The NI column reports unadjusted non-inferiority outcomes; $\Delta\%$ is the relative regret gap to SPO+.

Configuration	$\lvert\mathcal{P}\rvert$	Coverage	Hit rate	Cosine	$\Delta\%$	NI
SP 5 base	70	1.00	1.000	**1.000**	-0.12	yes
Assign 8 aug	19415	1.00	0.541	**0.915**	-0.13	yes
SP 10 aug	7917	1.05	0.157	**0.850**	-0.02	yes
SP 10 base	977	1.01	0.029	**0.776**	+1.33	no
Assign 8 base	982	0.88	0.023	**0.755**	+1.97	no
SP 20 aug	17975	0.64	0.000	**0.643**	+0.78	no
SP 20 base	1000	0.33	0.000	**0.607**	+2.61	no
Knap 20 aug	13316	0.88	0.000	**0.226**	+4.96	no
Knap 20 base	745	0.81	0.000	**0.181**	+40.88	no

6 Conclusion

PoolDFL constructs a fixed candidate pool from training optima and their feasible neighbours, and evaluates a restricted SPO+ objective through matrix operations. The objective recovers the full loss when the pool contains a global probe optimum. In the cross-task comparisons, neighbourhood augmentation lowers mean regret relative to the base pool. The gradient diagnostics link closer alignment with the full objective to smaller regret gaps.

USE OF LARGE LANGUAGE MODELS

Large language models were used for code implementation and language editing of the manuscript. The authors take full responsibility for the implementation, results and manuscript.

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.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, 2017.

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.

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

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with Fenchel–Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

Priya 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.

Othman El Balghiti, Adam N. Elmachtoub, Paul Grigas, and Ambuj Tewari. Generalization bounds in the predict-then-optimize framework. In *Advances in Neural Information Processing Systems*, volume 32, pp. 14389–14398, 2019.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1): 9–26, 2022.

Aaron Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. MIPaaL: Mixed integer program as a layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1504–1511, 2020. doi: 10.1609/aaai.v34i02.5509.

Nam Ho-Nguyen and Fatma Kılınç-Karzan. Risk guarantees for end-to-end prediction and optimization processes. *Management Science*, 68(12):8680–8698, 2022. doi: 10.1287/mnsc.2022.4321.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Junhyeong Lee, Sangjin Jin, and Yongjae Lee. Decision-focused learning via tangent-space projection of prediction error. *arXiv preprint arXiv:2605.01361*, 2026.

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

Jayanta Mandi and Tias Guns. Interior point solving for LP-based prediction+optimisation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7272–7282, 2020.

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.

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

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

Jayanta Mandi, Ali İrfan Mahmutoğulları, Senne Berden, and Tias Guns. Minimizing surrogate losses for decision-focused learning using differentiable optimization. In *European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 3888–3895. IOS Press, 2025. doi: 10.3233/FAIA251273.

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, 2021. doi: 10.24963/ijcai.2021/390.

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

Anselm Paulus, Michal Rolínek, Vít Musil, Brandon Amos, and Georg Martius. CombOptNet: Fit the right NP-hard problem by learning integer programming constraints. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8443–8453, 2021.

Paula Rodriguez-Diaz, Kirk Bansak, and Elisabeth Paulson. A dual perspective on decision-focused learning: Scalable training via dual-guided surrogates. *arXiv preprint arXiv:2511.04909*, 2025.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025. doi: 10.1016/j.ejor.2024.03.020.

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, pp. 1320–1332, 2022. doi: 10.52202/068431-0097.

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, 2023.

Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 2024a. doi: 10.1007/s12532-024-00255-x.

Bo Tang and Elias B. Khalil. CaVE: A cone-aligned approach for fast predict-then-optimize with binary linear programs. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 193–210, 2024b.

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

Beichen Wan and Mo Liu. A solver-free training method for predict-then-optimize. In *International Conference on Machine Learning*, 2026.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658.

A PROOFS

A.1 PROPOSITION 1

Write $A = c - 2\hat{c}$. For any $\mathcal{P} \subseteq \mathcal{V}(S)$,

$$\ell_{\mathcal{P}}(\hat{c}, c) = \Big(\max_{w \in \mathcal{P}} A^\top (w - w^\star) \Big)_+ = \max_{w \in \mathcal{P} \cup \{w^\star\}} A^\top w + 2\hat{c}^\top w^\star - c^\top w^\star,$$

because the term $w = w^\star$ contributes $A^\top(w^\star - w^\star) = 0$, which is exactly what $(\cdot)_+$ adds, and $A^\top w^\star + 2\hat{c}^\top w^\star - c^\top w^\star = 0$.

Convexity. Each $\hat{c} \mapsto A^\top(w - w^\star)$ is affine, a maximum of affine functions is convex, and $(\cdot)_+$ is convex and non-decreasing.

Bound. Maximising over $\mathcal{P} \cup \{w^\star\} \subseteq S$ cannot exceed the maximum over S, so $\ell_{\mathcal{P}} \leq \ell_{\text{SPO+}}$ for every prediction.

Equality. $w^\star(2\hat{c} - c)$ maximises $A^\top w = (c - 2\hat{c})^\top w$ over S. If it lies in $\mathcal{P}$, the two maxima coincide.

Subgradient. With $w^\dagger$ the maximiser, Danskin's theorem gives $\partial_{\hat{c}} \ell_{\mathcal{P}} \ni -2w^\dagger + 2w^\star = 2(w^\star - w^\dagger)$ whenever the bracket is positive; zero is a valid choice on the nonpositive hinge branch. Equivalently, choose a maximiser over $\mathcal{P} \cup \{w^\star\}$, taking the label when selecting the zero subgradient. Under the equality condition, matching the maximiser and zero-branch choice of SPO+ yields the same subgradient. □

A.2 THE MARGIN POWER IN THE ADJACENT-FACET HESSIAN

For $\mathcal{R}(\mu) = \sum_j m_j \Phi(g_j(\mu))$ with $g_j(\mu) = (-m_j - v_j^\top \mu)/(\sigma\|v_j\|)$ and $z_j = m_j/(\sigma\|v_j\|)$, using $\varphi'(t) = -t\varphi(t)$ and $\nabla g_j = -v_j/(\sigma\|v_j\|)$,

$$\nabla_\mu^2 \mathcal{R}\big|_{\mu=0} = \sum_j m_j \, \varphi'(g_j) \, (\nabla g_j)^{\otimes 2} = \sigma^{-3} \sum_j \frac{m_j^2 \, \varphi(z_j)}{\|v_j\|^3} \, v_j v_j^\top \succeq 0.$$

In the large-noise limit $z_j \to 0$ only $\varphi(z_j) \to \varphi(0)$ becomes constant; z_j is itself proportional to m_j. The limiting weight is therefore m_j^2. The σ^{-3} prefactor makes the Hessian vanish as $\sigma \to \infty$, which a version with weight m_j obtained by holding $z_j\varphi(z_j)$ constant would not.

Table 2 compares both margin powers under the same protocol and neighbourhood.

B THE CONSTRUCTED MOTIVATING EXAMPLE

Figure 1 uses the six right-down paths on a 3×3 grid from the top-left to the bottom-right node. R denotes a right move and D a down move. Set true edge costs to 1 on RRDD and 1.5 elsewhere. Set the probe costs q to 2 on RRDD, 1 on DDRR and 5 on the remaining four edges, and define $\hat{c} = (c + q)/2$. The complete enumeration is:

Path	$c^\top w$	$q^\top w$	$\hat{c}^\top w$
RRDD	4.0	8	6.00
RDRD	5.0	14	9.50
RDDR	5.5	13	9.25
DRRD	5.5	13	9.25
DRDR	6.0	12	9.00
DDRR	6.0	4	5.00

RRDD is the unique true optimum, and its only unit-square swap is RDRD. Thus minimising $q^\top w$ over the label and this implemented neighbour returns RRDD, with restricted loss $8 - 8 = 0$ and zero gradient. Both the full probe solve and the predicted-cost solve uniquely return DDRR. The full SPO+ loss is $8 - 4 = 4$, while the unnormalised decision regret is $6 - 4 = 2$. The pool $\{\text{RRDD}, \text{DDRR}\}$ recovers that loss and its nonzero subgradient. To realise DDRR as another training label, assign its four edges cost 1 and all other edges cost 3; DDRR is then uniquely optimal. The construction uses the unit-square neighbourhood, a subset of the full LP adjacency relation.

C Implementation Details

Grid shortest path. $N \times N$ right-down DAG, $n = 2N(N-1)$ edges, all s–t paths of equal length $2(N-1)$, so a uniform shift of the costs never changes the argmin. The unit-square enumeration in Section 4 visits at most $2N-3$ swappable corners per path; the measured mean is close to $N-1.5$ (Table 6 reports $K = 3.7, 8.2, 13.2, 16.4$ at $N = 5, 10, 15, 20$). This unit-square-swap relation is a subset of true LP adjacency; Table 7 measures the resulting coverage gap.

Assignment and knapsack. 8×8 assignment uses a Hungarian oracle with 4-cycle adjacency. Knapsack uses 20 items, two capacity constraints, an exact solver and a heuristic single-item-swap neighbourhood.

Baseline losses. With $\sigma = +1$ for minimisation and -1 for maximisation, $s_j = \sigma v_j^\top \hat{c}$ and $e = \hat{c} - c$:

$$
\begin{aligned}
\ell_{\text{LAVA}} &= \textstyle\sum_j \max(-s_j, -\epsilon), && \epsilon = 0.1 \text{ (Berden et al., 2025)},\\
\ell_{\text{WISE}} &= \|c\| \left\| \hat{c} - c/\|c\| \right\|^2 && \text{(Wan \& Liu, 2026)},\\
\ell_{\text{PEAR}} &= \|Pe\|^2 + \beta \tfrac{\|Pe\|}{\|(I-P)e\|} \|(I-P)e\|^2, && \beta = 0.1 \text{ (Lee et al., 2026)},\\
\ell_{\text{CAVE}} &= 1 - \cos\big(\hat{c},\ \Pi_{\mathcal{C}}(\hat{c})\big), && \mathcal{C} = \{x : \sigma v_j^\top x \geq 0\ \forall j\} \text{ (Tang \& Khalil, 2024b)},\\
\ell_{\text{RHL}} &= \textstyle\sum_j m_j^p\, (v_j^\top e)^2, && p \in \{1, 2\},\\
\ell_{\text{ARM}} &= \textstyle\sum_j m_j \big[(m_j - s_j)_+\big]^2.
\end{aligned}
$$

In our PEAR implementation, P is fixed at $w^\star(c)$ and projects orthogonally onto $\operatorname{span}\{v_j\}$.

Baseline settings and sensitivity. The main comparisons use the published configurations above. Table 2 also reports LAVA at $\epsilon = 0$ and PEAR at $\beta = 0$ to show parameter sensitivity. For LAVA, changing to $\epsilon = 0.1$ gives regret $0.1818 \to 0.1822$ on this benchmark; Berden et al. (2025) report stronger threshold sensitivity. At $\epsilon = 0$, the loss is also minimised by $\hat{c} = 0$. For PEAR, normal injection improves regret from 0.1596 to 0.1388. The WISE implementation uses target normalisation and weight $\|c\|$. We compute CAVE's $\Pi_{\mathcal{C}}$ via Moreau's decomposition, $\Pi_{\mathcal{C}}(x) = x + V^\top \mu^\star$ with $\mu^\star = \arg\min_{\mu \geq 0} \|V^\top \mu + x\|$, a K-dimensional non-negative least-squares problem requiring no LP solve.

Pool-size ablation. For 10×10 shortest path with $n_{\text{train}} = 1000$, the full base pool contains 970 distinct vertices. Subsets are drawn uniformly at random and nested.

D Tie-breaking in the Full-Pool Identity Check

On 5×5 the base pool contains all 70 vertices of the path polytope, so Proposition 1 makes PoolDFL and SPO+ the same loss. The measured paired regret difference is -0.12%. Running both arms with identical seeds, data order and initialisation and comparing per-step, the loss values agree throughout and the gradients agree exactly for the first 695 optimiser steps. At step 696 the probe cost $c - 2\hat{c}$ admits two maximisers of equal objective; PoolDFL takes a dense float32 $\arg\max$ over the pool and SPO+ calls a float64 Dijkstra, and at a tie they return different vertices. Both selected subgradients are valid, and the trajectories diverge from that step. After 150 epochs, their final regrets differ by $\sim 1.5 \times 10^{-3}$. Thus equal loss functions can produce distinct valid trajectories when maximiser tie-breaking differs.

E COVERAGE MEASUREMENT

Table 5 compares implemented neighbourhoods with true adjacency, and Table 6 reports coverage across tasks and perturbation scales. Table 7 separates the aggregation and coverage effects.

Table 5: Implemented neighbourhoods and true LP adjacency. The true neighbourhood contains 11–574 times as many vertices over these problems. Vertex counts are per instance; $|V|$ denotes the full polytope vertex set.

Problem	$\lvert V\rvert$	K implemented	K true	σ	Cov. implemented	Cov. true
SP 5×5	70	3.4	38	0.5	0.36	0.92
				7.0	0.33	0.85
SP 6×6	252	4.2	104	0.5	0.35	0.85
				7.0	0.23	0.87
SP 7×7	924	5.6	309	0.5	0.34	0.85
				7.0	0.22	0.86
Assignment 8×8	40320	28.0	16064	0.5	0.46	0.93
				7.1	0.27	0.99

For each of 40 costs (30 for the cross-problem table) we draw 100–160 perturbations $\delta \sim \mathcal{N}(0, \sigma^2 I)$, solve exactly at $c + \delta$ for the ground truth, and evaluate the true-cost regret of the vertex that a maximum over the candidate set selects at $c + \delta$. Coverage is the ratio of the two means. Values slightly above one occur because a restricted set can select a vertex whose true cost is worse than that of the unrestricted optimum; coverage near one means the restricted decision has the same expected regret as the unrestricted one.

The residual σ of Section 5.4 is the standard deviation of $\hat{c} - c$ on held-out data for a linear predictor trained to convergence with mean squared error under the same protocol.

Table 6: Expected-regret ratios for each candidate set. The left block uses DGP noise $\sigma = 0.5$; the right block uses the measured residual scale of a trained linear predictor. Here, 1-hop denotes the implemented neighbourhood, pool the training optima, and aug their 1-hop expansion. Pool construction reuses labels and adds no solver calls. Intervals are bootstrap 95% confidence intervals over costs, reported as half-widths.

				$\sigma = 0.5$				$\sigma =$ residual		
Problem	n	K	$\lvert\mathcal{P}\rvert$	1-hop	pool	aug	σ	1-hop	pool	aug
SP 5×5	40	3.7	70	0.41 ±0.06	1.00 ±0.00	1.00 ±0.00	7.2	0.33 ±0.04	1.00 ±0.00	1.00 ±0.00
SP 10×10	180	8.2	970	0.22 ±0.02	1.01 ±0.06	1.05 ±0.03	6.9	0.16 ±0.01	1.01 ±0.03	1.00 ±0.01
SP 15×15	420	13.2	999	0.17 ±0.01	0.71 ±0.07	1.02 ±0.05	6.9	0.10 ±0.01	1.03 ±0.03	1.02 ±0.02
SP 20×20	760	16.4	1000	0.12 ±0.01	0.33 ±0.05	0.64 ±0.06	6.8	0.08 ±0.00	1.03 ±0.03	1.02 ±0.02
Assignment 8×8	64	28.0	951	0.45 ±0.04	0.88 ±0.10	1.00 ±0.03	7.1	0.27 ±0.01	0.99 ±0.05	0.99 ±0.01
Knapsack (20)	20	18.0	580	0.21 ±0.06	0.81 ±0.04	0.88 ±0.04	7.5	0.32 ±0.04	0.71 ±0.02	0.87 ±0.01

Table 7: Aggregation and coverage components of the adjacent-facet model. Entries are ratios to true expected regret. The sum model adds facet tails as in (4); sel evaluates the vertex selected under the perturbed cost; walk runs steepest descent over the 1-hop neighbourhood to local optimality. Aggregation error is sum minus sel on the same neighbourhood; coverage error is one minus walk. Grid shortest path, 40 costs × 160 draws, $\sigma = 0.5$.

			1-hop		2-hop			agg. error		cov. error
N	K	K_2	sum	sel	sum	sel	walk	1-hop	2-hop	walk
5	3.6	10.7	0.52	0.40	1.57	0.62	0.47	0.11	0.94	0.53
10	8.1	43.0	0.46	0.23	2.68	0.38	0.40	0.23	2.30	0.60
15	13.3	105.9	0.45	0.17	3.84	0.29	0.40	0.28	3.55	0.60
20	16.1	150.5	0.40	0.12	4.12	0.20	0.35	0.28	3.92	0.65

F ADJACENT-FACET LOSS CONFIGURATIONS

Table 8: Pre-registered screen of 17 adjacent-facet configurations on 10×10 shortest path, five seeds, against SPO+ at 0.1265. The registered design varies one loss component at a time from two published members of the family and includes 2-hop variants and warm-started pivot walks. Configuration names follow `threshold_aggregation[_weight][_power]`.

Configuration	Regret	vs SPO+	Wins
`arm_as_max`	0.1414	$+11.71\%$	0/5
`arm_full_sum_margin_p2`	0.1459	$+15.31\%$	0/5
`half_max_margin_p2`	0.1476	$+16.67\%$	0/5
`half_sum_ext`	0.1478	$+16.78\%$	0/5
`full_max`	0.1478	$+16.83\%$	0/5
`half_max_margin`	0.1480	$+17.00\%$	0/5
`half_sum`	0.1505	$+18.90\%$	0/5
`half_max`	0.1518	$+19.97\%$	0/5
`pivot_k1`	0.1520	$+20.16\%$	0/5
`half_lse`	0.1531	$+20.95\%$	0/5
`half_max_p2`	0.1531	$+20.99\%$	0/5
`half_max_ext`	0.1533	$+21.11\%$	0/5
`pivot_k3`	0.1557	$+23.04\%$	0/5
`half_mean`	0.1557	$+23.07\%$	0/5
`zero_max`	0.1821	$+43.88\%$	0/5
`lava_eps_sum`	0.1822	$+44.00\%$	0/5
`eps_max`	0.1839	$+45.34\%$	0/5

Configurations are named `threshold_aggregation[_weight][_power]`, with `ext` marking the 2-hop neighbourhood and `pivot_k` a warm-started steepest-descent walk of k steps at the SPO+ probe cost. The pre-registered screen required five-seed mean regret no higher than SPO+ before 12-seed confirmation. Confirmatory and transfer comparisons required a significant one-sided paired improvement.

Replacing summation with selection in ARM gives `arm_as_max` and reduces the regret gap from $+15.3\%$ to $+11.7\%$. The `half_max` configuration is SPO+ restricted to the 1-hop neighbourhood, corresponding to Proposition 1 with $\mathcal{P}$ set to the adjacent vertices. Its regret gap is $+20.0\%$; the training-label pool uses the same loss with a larger candidate set.

G COMPARISON AND DIAGNOSTIC TABLES

Table 9 reports the statistical tests behind Figure 3. It includes the full non-inferiority family; its 12-seed SP 10 row is the pilot preceding the 35-seed confirmation and is excluded from the nine-test Holm family.

H ADDITIONAL ABLATIONS

Table 10 is the grid-size sweep with the base pool and Table 11 the pool-size sweep at $N=10$. The pool-size rows report point estimates averaged over five seeds.

Table 9: Non-inferiority of POOLDFL against SPO+, pre-registered at a 1% margin, with the pool's measured coverage from Table 6 alongside. Positive $\Delta\%$ means worse than SPO+. The SP 10×10 base-pool rows distinguish the 12-seed pilot from the 35-seed confirmation. The augmented pool was selected on knapsack and assignment and then tested once on 10×10. The Non-inf. column reports unadjusted test outcomes. Holm correction over the nine tests excluding the pilot retains the 5×5 identity and augmented assignment; augmented SP 10×10 meets the unadjusted criterion. Seed counts refer to regret comparisons; the SP 10×10 confirmation uses timings from 23 seeds for SPO+ and the base pool, and 35 for the augmented pool.

Problem	Cov.	Seeds	$\Delta\%$	95% CI of paired diff.	p_{ni}	Non-inf.	Speed-up
Base pool: the training optima							
SP 10×10 (pilot)	1.01	12	-0.08	[-0.0027, +0.0025]	0.123	no	$15.0\times$
SP 10×10	1.01	35	+1.33	[+0.0003, +0.0031]	0.727	no	$8.8\times$
SP 5×5 (identity, see text)	1.00	12	-0.12	[-0.0005, +0.0001]	2.26e-08	**yes**	$3.7\times$
Knapsack (20)	0.81	5	+40.88	[+0.0292, +0.0446]	1	no	$14.8\times$
Assignment 8×8	0.88	12	+1.97	[+0.0019, +0.0075]	0.953	no	$1.3\times$
SP 10×10, MLP predictor	1.01	5	+0.39	[-0.0014, +0.0021]	0.235	no	–
Augmented pool: + the 1-hop neighbours of pool members							
SP 10×10	1.05	35	-0.02	[-0.0014, +0.0013]	0.0285	**yes**	$3.7\times$
Assignment 8×8	1.00	35	+0.16	[-0.0007, +0.0014]	0.000246	**yes**	$0.5\times$
Knapsack (20)	0.88	5	+4.96	[+0.0014, +0.0076]	0.983	no	$7.9\times$
SP 20×20	0.64	5	+0.78	[-0.0021, +0.0038]	0.414	no	$4.1\times$

Table 10: Grid-size sweep with the base pool, 12 seeds. Coverage is the pool's share of true expected regret at $\sigma{=}0.5$ from Table 6, measured without any training.

N	Pool coverage	SPO+	POOLDFL	$\Delta\%$	p_{ni}	Speed-up
5	1.00	0.1536	0.1534	-0.12	2.26e-08	$3.7\times$
10	1.01	0.1328	0.1327	-0.08	0.123	$9.5\times$
15	0.71	0.1158	0.1187	+2.42	0.896	$11.6\times$
20	0.33	0.1162	0.1194	+2.76	1	$12.1\times$

Table 11: Base-pool size, measured coverage and decision quality on 10×10, 5 seeds. Subsets are drawn at random and nested; entries report mean regret and its relative gap to SPO+.

$\lvert\mathcal{P}\rvert$	Coverage	Regret	vs SPO+
8	0.18	0.1428	$+12.84\%$
32	0.43	0.1439	$+13.70\%$
128	0.72	0.1424	$+12.56\%$
512	0.94	0.1406	$+11.10\%$
all 970	1.02	0.1266	$+0.02\%$

Table 12: Candidate-pool and aggregation ablations on 10×10 shortest path, 12 seeds paired with SPO+ (mean regret 0.1328). The summed and averaged variants have regret gaps of 0.9–1.0%, with paired confidence intervals spanning zero. Masking a sample's own optimum removes a zero loss term.

Pool and aggregation	Regret	Std	vs SPO+	95% CI of paired diff.	p
POOLDFL, base pool ($\lvert\mathcal{P}\rvert$=970)	0.1327	0.0112	-0.08%	[-0.0027, +0.0025]	0.9282
POOLDFL, augmented pool ($\lvert\mathcal{P}\rvert$=7917)	0.1316	0.0116	-0.87%	[-0.0039, +0.0016]	0.3687
base pool, own optimum masked out	0.1327	0.0112	-0.08%	[-0.0027, +0.0025]	0.9282
base pool, summed instead of selected	0.1340	0.0100	$+0.91\%$	[-0.0026, +0.0051]	0.5049
base pool, averaged instead of selected	0.1341	0.0102	$+0.97\%$	[-0.0025, +0.0051]	0.4661
random feasible vertices, $\lvert\mathcal{P}\rvert$=970	0.1494	0.0132	$+12.55\%$	[+0.0108, +0.0226]	0.0001
random feasible vertices, $\lvert\mathcal{P}\rvert$=7900	0.1441	0.0133	$+8.50\%$	[+0.0089, +0.0136]	0.0000

Table 13: Effect of candidate provenance on 10×10 shortest path, 12 seeds paired with SPO+ (mean regret 0.1328).

Pool provenance	Regret	vs SPO+	95% CI of paired diff.	p
Training optima, all 1000 instances	0.1327	-0.08%	[-0.0027, +0.0025]	0.9282
augmented	0.1316	-0.87%	[-0.0039, +0.0016]	0.3687
Training optima, a random half	0.1461	$+10.06\%$	[+0.0081, +0.0186]	0.0002
Held-out optima, 500 instances, same DGP	0.1528	$+15.09\%$	[+0.0146, +0.0255]	0.0000
augmented	0.1500	$+12.96\%$	[+0.0107, +0.0238]	0.0001

I EVALUATION CRITERIA

Configuration lists and evaluation criteria were specified before the corresponding runs. The five-seed screen required mean regret no higher than SPO+ before a configuration advanced to paired superiority tests. Tables 8 and 11 report the local configurations and the base-pool comparison, respectively. Superiority required a significant one-sided paired improvement. The grid-size criterion required no significantly positive Spearman correlation between grid size and relative regret gap at $\alpha = 0.05$; Table 10 reports the size sweep.

For non-inferiority, the margin is 1% of mean SPO+ regret and the one-sided paired test uses $\alpha = 0.05$. Table 9 gives the regret differences, two-sided 95% intervals and unadjusted test results. The nine-test Holm family excludes the 12-seed shortest-path pilot; Section 5.2 and Figure 3 report the corrected results.

J WARM-STARTED PIVOT WALKS

Unit-square descent from the cached $w^\star(c)$ at probe cost $2\hat{c} - c$ recovers the full SPO+ objective on 54% of instances at N=10 under a small perturbation; the corresponding rate is near zero at N=20 under a large perturbation. Recovery stabilises after two or three moves; gradient cosine with SPO+ plateaus at 0.43 for N=10. Training regret gaps are $+20.2\%$ for `pivot_k1` and $+23.0\%$ for `pivot_k3` (Table 8).