NATIONAL-SCALE MANURE LIFE-CYCLE OPTIMIZATION WITH LEARNED PRICES AND OPTIMALITY BOUNDS

Anonymous authors
Paper under double-blind review

ABSTRACT

National cattle-manure life-cycle optimization coordinates processing facilities, transport and crop application across roughly 33 billion potential transport variables. Restricting routes makes planning tractable but leaves the quality of the resulting plan unknown relative to the full model. We compute a feasible plan on a sparse network and bound the full optimum through a spatial relaxation that preserves local balances, transport and integer facility decisions. Lagrangian multipliers price shipments by material and destination, allowing independent regional solves whose upper bounds sum to a valid national bound. We learn eight material-specific adjustments to linear-programming dual prices from regional subproblem feedback and use them to initialize convex multiplier refinement. On 63,873 road-covered cells of the contiguous US, our approach produces a feasible $129.870B/year plan, 4.00% above the best graph sampling and coarsening plan in our tests ($124.873B/year), and a $132.154B/year upper bound, leaving a 1.73% gap. This bound improves on the full continuous relaxation, whose optimum is at least $135.110B/year. On sixteen 256-cell regions from the same US data, learned initialization improves bounds in 15 cases after twelve rounds of the same optimizer, by a mean $0.570M/year per region relative to initialization without learned adjustments. Our upper bounds are also $9.018M/year lower on average than those from a recent learning-based approach to Lagrangian multiplier prediction.

1 INTRODUCTION

Manure's nutrient value depends on where it is produced and used. National life-cycle optimization (LCO) must therefore coordinate processing facilities, transport and crop application. The reference model tracks eight materials: raw manure; digestate, the product of anaerobic digestion; the separated solid and liquid fractions of each; and biochar from each solid fraction. On 63,873 road-covered 10 km grid cells of the contiguous US, these materials imply roughly 33 billion potential transport variables. A compact explicit-array representation of those variables alone requires about 1.83 TB, before solver working memory, exceeding our largest node's nominal 1.5 TB RAM (Appendix E). We seek a feasible national plan and a bound on its distance from optimality without constructing this dense mixed-integer program (MIP).

Restricting routes makes planning tractable, but a small restricted-network solver gap does not measure the profit available through omitted routes. An upper bound must cover the full transport model. Graph sampling and coarsening (gSC) addresses this need by aggregating material balances and using minimum freight costs between groups (Ma & Zavala, 2022). Our spatial relaxation instead preserves cell-level balances, transport within each block and integer facility decisions. It relaxes the constraints matching shipments between blocks, using Lagrangian multipliers interpreted as per-tonne credits and charges for material delivered to a destination. These terms cancel when shipments match, so summing the regional upper bounds bounds the national optimum for any multiplier field.

Learning improves the choice of multipliers and hence the bound's tightness. We retain the geographic variation of linear-programming (LP) dual prices and learn one shared adjustment for each material. Regional export/import imbalances provide training feedback, requiring neither target mul-

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.

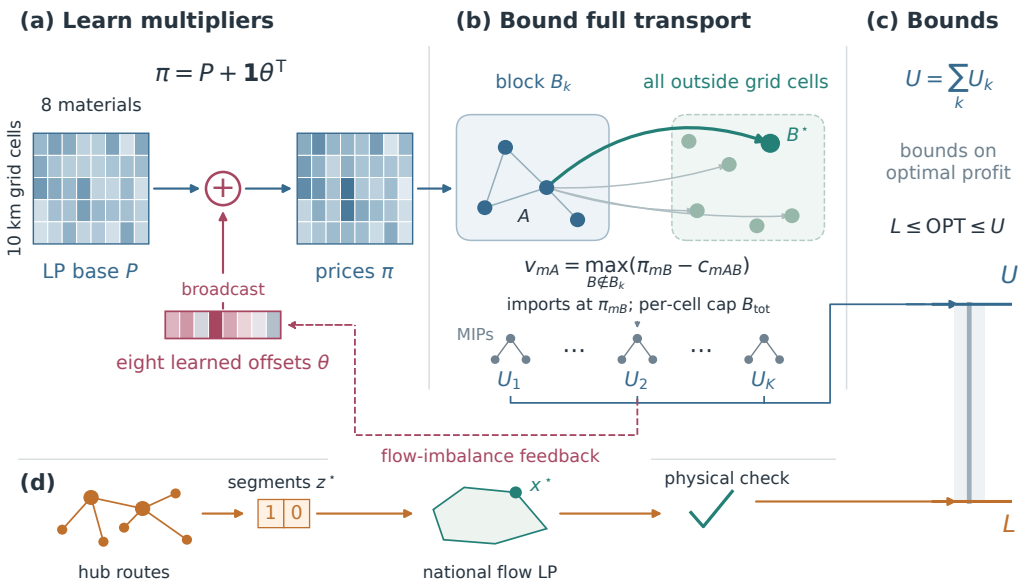


Figure 1: **Compact multiplier learning and spatial upper bounds.** (a) Eight learned material offsets are shared across LP-price rows, each representing a 10 km grid cell. (b) Block B_k prices exports against every outside grid cell. Flow imbalances train the offsets and support refinement. (c) Summed solver bounds enclose the optimum with (d) a checked sparse plan. National deployment also applies a material-price floor. The matrices, networks and LP geometry are schematic.

tipliers nor a solved dense national instance. The eight parameters can be reused at national scale, while the instance-specific LP base supplies local economic information. The bound is convex in these adjustments, allowing further optimization from the learned initialization. Figure 1 connects this learned bound computation to a separate sparse-network planner.

Our contributions are:

- **A spatial relaxation covering all transport routes.** We preserve local LCO structure while replacing explicit cross-block route copies with aggregate imports and exports. Under the stated incoming-flow assumptions, destination multipliers attain the best route-multiplier bound.
- **Eight transferable learned adjustments.** The representation preserves geographic price differences, learns from integer-subproblem feedback and supports convex refinement without growing the learned parameter count with the number of locations.
- **National results and controlled learning evidence.** We obtain a \$129.870B/year plan, 4.00% above the best tested gSC plan, with a 1.73% optimality gap. A frozen regional test isolates the gain from learned initialization; a separate comparison evaluates the complete method against a published multiplier predictor.

2 Related Work

Manure-management models study regional processing networks (Sharara et al., 2017; Sampat et al., 2019; Hu et al., 2019), national nutrient transport (Akram et al., 2019) and livestock relocation (Yin et al., 2025). Recent work considers consequential life-cycle accounting (Preuss & You, 2024) and robust collection and routing with accelerated Benders decomposition (Chen et al., 2026). Their physical objectives differ, so algorithmic comparisons here fix the same deterministic LCO formulation.

The closest planning baseline, gSC (Ma & Zavala, 2022), already pairs sparse feasible plans with full-model bounds and retains integer technologies. Its aggregation removes individual-cell bal-

ance constraints and within-group freight. Our relaxation preserves both and prices cross-block shipment consistency; this is the structural distinction. For learning, Demelas et al. (2024) predict Lagrangian multipliers from a MIP's variable–constraint graph and LP features by optimizing the resulting bound. Le et al. (2026) analyze multiplier prediction and warm starts, while Bessa et al. (2025) learn valid dual bounds in constraint programming. Bound-driven learning is an existing principle. Our contribution is the spatial relaxation and compact multiplier representation that apply it without constructing the national MIP graph. Appendix H gives the broader context.

3 Model and method

Let $\mathcal{C}$ be the grid cells, $n = |\mathcal{C}|$, $\mathcal{M}$ the eight materials, $\mathcal{O}$ the digestion, separation and pyrolysis technologies, and $\mathcal{P}$ the crop types. Indices $l, A, B \in \mathcal{C}$, $m \in \mathcal{M}$, $o \in \mathcal{O}$ and $p \in \mathcal{P}$ identify their elements. Cell l supplies b_l tonnes of raw manure per year. Nonnegative variables x_{mAB} ship material m from A to $B \neq A$. Local continuous decisions are processing C_{mlo}, storage s_{ml} and crop and other local application a_{mlp}, a^0_{ml}, all in t/yr; yield fraction u_{lp} and biochar bonus w_{lp}; capacity κ_{lo} in t/yr; and capital cost χ_{lo} in dollars. Appendix A gives their physical constraints, with the complete notation in Table 6.

Material balances link transport, processing and local use. Conversion changes material type according to inventory yields; storage incurs material-specific losses before application. Crop yields follow capped responses to supplied nitrogen, phosphorus and potassium, with a separate biochar bonus. Nutrient shortfalls are allowed and affect yield, rather than making manure alone satisfy every recommendation. The model also permits application outside the modeled crop rows. These local relationships are preserved in every block and in the common physical formulation used for the published baselines.

Digestion and pyrolysis have concave piecewise-linear capital costs with a breakpoint K_o^{mid} and maximum capacity $K_o^{\max}$. The binary $z_{lo} = 0$ selects $0 \leq \kappa_{lo} \leq K_o^{\mathrm{mid}}$; $z_{lo} = 1$ selects $K_o^{\mathrm{mid}} \leq \kappa_{lo} \leq K_o^{\max}$. Capacity varies continuously within the selected segment. Separation has linear capital cost, so $z \in \{0,1\}^{n_z}$ with $n_z = 2n$: one binary per cell for each technology with a cost breakpoint. Fixing z leaves an LP.

Freight per tonne is $c_{mAB} = \phi_m + \nu_m d_{AB}$ for $A \neq B$, where d_{AB} is road distance and ϕ_m, ν_m are the fixed and per-kilometer rates; set $c_{mAA} = 0$ for price processing below. The objective is annual material and crop revenue minus processing, transport, storage, application and annualized capital costs. Cells couple only through material flows. Let OPT denote the unknown optimum with all routes allowed.

3.1 Sparse planning and a lower bound

We first optimize facility segments on sparse local networks. Nationally, 64 planning blocks of about 1,000 cells are divided into clusters, with a representative cell serving as each cluster's computational hub. Cells connect to their hub and hubs to each other. Time-limited MIPs on these networks supply feasible segment choices $z^\star$. We fix $z = z^\star$ and solve one national LP on the 16 nearest road neighbors plus all routes retained by those plans. It reoptimizes continuous flows, processing and capacities within the selected segments, including trade across blocks.

Proposition 1 (Feasible plan)**.** *The national LP is feasible; its solution is feasible for the dense program. Its checked value L satisfies $L \leq \mathrm{OPT}$.*

The concatenated local plans provide a feasible start. Setting omitted flows to zero embeds every restricted solution in the dense model (Appendix I).

3.2 Spatial relaxation and a full-model upper bound

For the upper bound, use a separate partition into disjoint blocks $B_k \subseteq \mathcal{C}$, $k = 1, \ldots, K$, whose union is $\mathcal{C}$. Nationally, $K = 2{,}048$ and each block has at most 32 cells. Split each cross-block shipment into sender and receiver copies and relax their equality. A multiplier $\pi_{mB} \in \mathbb{R}$, in \$/t, gives the sender a credit and the receiver a charge for material m delivered to cell B. These are optimization parameters, fixed during each block solve. Matching shipments have canceling payments.

Each origin may export to any outside destination. Its best credit after freight, or export *netback*, is

$$v_{mA}(\pi) = \max_{B \notin B_k} \{\pi_{mB} - c_{mAB}\}, \qquad A \in B_k. \tag{1}$$

Scanning all destinations gives one coefficient per origin and material, without an explicit variable for every outside route. Let $x^{\uparrow}_{mA}, x^{\downarrow}_{mB} \geq 0$ denote total exports and imports in t/yr. Let ξ_k collect these quantities, internal flows and local physical decisions. Its feasible set $\mathcal{X}_k$ imposes variable domains, cell-level balances, conversion, crop response and facility constraints from Appendix A. Write $F_k(\xi_k)$ for local annual revenue minus local costs and internal freight, and $B_{\text{tot}} = \sum_l b_l$ for total instance supply. The block's exact optimum is

$$\mathcal{L}_k(\pi) = \max_{\xi_k \in \mathcal{X}_k} F_k(\xi_k) + \sum_{m \in \mathcal{M},\, A \in B_k} v_{mA}(\pi) x^{\uparrow}_{mA} - \sum_{m \in \mathcal{M},\, B \in B_k} \pi_{mB} x^{\downarrow}_{mB} \tag{2}$$
$$\text{s.t. } x^{\downarrow}_{mB} \leq B_{\text{tot}} \quad (m \in \mathcal{M},\ B \in B_k).$$

The cap keeps independently chosen imports finite. The model's conversion graph is acyclic and no operation increases total output mass. After removing positive-cost transport cycles, some dense optimum satisfies these per-cell, per-material caps. This remains true when a route leaves and re-enters a block on the supplied nonmetric road network; an aggregate block cap would not have that guarantee (Appendix I).

Proposition 2 (Validity)**.** *If every block includes all internal routes and all outside export destinations, then for every multiplier field π,*

$$U(\pi) = \sum_{k=1}^{K} \mathcal{L}_k(\pi) \geq \text{OPT}. \tag{3}$$

A valid solver upper bound $U_k(\pi) \geq \mathcal{L}_k(\pi)$ may replace an unfinished block optimum.

To see why, split a dense optimum into sender and receiver copies. Their payments cancel; choosing the best destination and optimizing each block can only increase its value. The computed bound is $\sum_k U_k(\pi)$, using every block's returned bound, including at time limits.

Why destination multipliers suffice. Using a separate multiplier for every route would reintroduce a quadratic representation. The next result shows that grouping by destination can preserve the best attainable bound for this split formulation.

Proposition 3 (Destination prices suffice)**.** *For $K \geq 2$ nonempty blocks, let $E_\times$ contain the admissible cross-block arcs, and let $U^{\text{arc}}(\mu)$ use one multiplier μ_{mAB} per such arc, with the per-cell import caps of Eq.*(2)*. Suppose every cell has an incoming arc in $E_\times$, and incoming copies are constrained only through their material totals at each receiving cell. Setting $\pi_{mB} = \min_{A:\,(A,B) \in E_\times} \mu_{mAB}$ gives $U(\pi) \leq U^{\text{arc}}(\mu)$. Consequently,* $\inf_\pi U(\pi) = \inf_\mu U^{\text{arc}}(\mu)$.

The receiving block can assign its imports to the cheapest incoming multiplier without changing its physical decisions. Every sender credit under μ is at least the corresponding destination price; the reverse embedding sets $\mu_{mAB} = \pi_{mB}$. Appendix I gives the full argument. This result concerns the entire $8n$-entry field; the learned eight-parameter class below is a further restriction.

3.3 Learning and refining material prices

LP duals encode spatial supply, demand and transport tradeoffs, but relax the facility binaries. We learn a correction for the integer block objective. Let π^{LP} be the duals of material balances in an LP on the chosen transport network. Sparse inputs omit some price inequalities, so we preprocess them with

$$T(\pi)_{mB} = \min_{A \in \mathcal{C}} \{\pi_{mA} + c_{mAB}\}. \tag{4}$$

This compares a destination's price to every origin's price plus direct freight. Bound validity holds for any processed field; repeated passes and the national material-price floor are specified in Appendices B.2 and G. The supplied road matrix can violate the triangle inequality, so one pass need not reach $T(\pi) = \pi$. The national CG base is processed to a checked fixed point; the regional pass

counts below remain fixed across the compared models. The upper-bound guarantee comes from the block relaxation for every field, independently of this preprocessing.

The floor uses a physical alternative to exporting: local storage and application outside the modeled crop rows. Let v_m^{loc} be this option's net value per tonne. When it is available without a quantity cap and freight is nonnegative, replacing π_{mB} by $\max\{\pi_{mB}, f_m\}$ for any $f_m \le v_m^{\mathrm{loc}}$ cannot increase the exact bound (Proposition 5). This provides a general material-based transformation; the regional initialization comparison uses the unfloored class.

For regional inputs, set $q = T(\pi^{\mathrm{LP}})$ and $P = T(q)$. We learn $\theta \in \mathbb{R}^8$, one adjustment in \$/t per material:

$$\pi_{mB} = P_{mB} + \theta_m. \tag{5}$$

The shared offset commutes with T, so $T(q + \theta) = P + \theta$. It preserves every geographic price difference $\pi_{mB} - \pi_{mA} = P_{mB} - P_{mA}$ and therefore every transport-price inequality satisfied by P. The same eight parameters are reused across instances; each instance retains its own LP base. Zero offsets recover the LP-derived control.

This separates two roles: the LP computes location-dependent economic information, while learning calibrates the relative values of materials for the integer-block objective. A cellwise predictor could change every price difference independently. The shared correction instead retains the base's spatial relationships and makes subsequent optimization eight-dimensional. The neural controls test whether a more expressive correction improves on this compact choice.

Learning from regional optimization. Choose a maximizing destination $B^\star_{mA}$ in Eq.(1). Optimized blocks supply the destination imbalance

$$g_{mB} = \sum_{k: B \notin B_k} \sum_{A \in B_k} \mathbf{1}[B = B^\star_{mA}] x^{\uparrow}_{mA} - x^{\downarrow}_{mB}. \tag{6}$$

The indicator assigns each export to its chosen destination. At optimal block solutions, g is a price subgradient, up to solver accuracy; $\sum_B g_{mB}$ is the subgradient for θ_m. At the current field, hold that solver feedback fixed and differentiate the surrogate

$$\ell(\theta; g, P) = \sum_{m,B} \mathrm{stopgrad}\left(\frac{g_{mB}}{\max\{\|g\|_1, 1\}}\right)\left(P_{mB} + \theta_m\right). \tag{7}$$

Here $\|g\|_1 = \sum_{m,B} |g_{mB}|$, and stopgrad prevents differentiation through the solver's discrete choices. The normalization scales the feedback before summing it into the eight parameter coordinates. We use Adam with learning rate 0.05 and clip parameter-gradient norms to five. Thus training needs no target-price labels or precomputed optimal multipliers. Thirty epochs on twelve regions give 360 updates; the final checkpoint is fixed independently of development scores. All training block solves reach optimal status. Feedback normalization, optimization settings and the physical-cell graph neural network (GNN) and multilayer perceptron (MLP) controls are detailed in Appendix B.

For a fixed base P, the export coefficient satisfies $v_{mA}(P + \theta) = v_{mA}(P) + \theta_m$.

Proposition 4 (Convex material-price calibration)**.** *The exact bound $F(\theta) = U(P + \theta)$ is convex in θ. Each feasible collection of block solutions supplies an affine lower cut, with slope equal to exports minus imports of each material.*

Each fixed block solution has an affine objective in θ; maximizing and summing gives the result. A proximal bundle method uses these cuts to choose successive offsets. Cuts require feasible incumbent objectives, while reported upper bounds use solver bounds. Appendix B.4 gives the cuts, proof and fixed optimizer settings. At evaluated offsets $\theta^{(t)}$, let a_t be the sum of feasible block-incumbent objectives and let $h_t \in \mathbb{R}^8$ collect their total material exports minus imports. The retained lower cut is

$$F(\theta) \ge a_t + h_t^\top(\theta - \theta^{(t)}). \tag{8}$$

The optimizer minimizes a model of these cuts with a quadratic penalty for departing from its accepted center. Replacing a_t with a solver upper bound would invalidate the lower-cut argument. Both initialization arms use the same cut construction, search box and update rules. Refinement uses the unfloored affine class; the national floor has a separate dominance argument.

Algorithm 1 Planning and bounds with learned material adjustments

Require: instance; planning/bound partitions; learned offsets θ; round budget $S \geq 1$
1: Solve sparse planner MIPs; fix their feasible segment choices $z^\star$
2: Solve the national flow LP; verify the plan and record profit L
3: Compute the LP-price base P; initialize the best bound $U \leftarrow +\infty$
4: **for** $s = 1, \ldots, S$ **do**
5: Set $\pi = P + \theta$; apply the declared floor for one-round national deployment
6: Solve all blocks at π; retain the smallest $U = \sum_k U_k(\pi)$ and its field
7: If another round remains, refine θ using available feasible-incumbent cuts
8: **return** the checked plan, best evaluated price field and interval $[L, U]$

One *round* solves every block once at fixed multipliers, possibly in parallel. The first round evaluates the initialization and counts toward the budget. Algorithm 1 summarizes deployment.

Computational scaling. A block with b cells retains $|\mathcal{M}|b(b-1)$ internal transport variables and $2|\mathcal{M}|b$ boundary quantities, without outside facility variables. At most 32 cells give at most 64 segment binaries per national bound subproblem. This reduces the integer optimization to small local problems while Eq.(1) still accounts for all outside destinations. Larger blocks retain more transport consistency and cost more to solve; a single block with no boundary variables recovers the dense model.

The destination scans still require $O(|\mathcal{M}|n^2)$ arithmetic, streamed in block-sized batches. The float32 national road cache occupies 15.2 GiB. Learning targets repeated block-solver evaluations; it does not remove that scan or the instance-specific LP input. The reported costs therefore separate input construction, offline training and online evaluation.

4 Experiments

We ask whether the method produces a useful national plan and bound, whether learning helps after subsequent optimization, and how the complete approach compares with published algorithms.

Data and evaluation. The contiguous-US domain uses a 10 km Equal Earth grid, GLW4 cattle densities, SPAM crop area/production (FAO, 2024; International Food Policy Research Institute, 2024) and January 2026 roads. Road filtering retains 63,873 of 77,945 cells and 91.2% of cattle. The supplied reference model uses 14.4 tonnes of manure per head per year, a $200/t biochar subsidy and no carbon-policy cost; yield response permits nutrient shortfalls. The physical formulation and parameter provenance are in Appendix A.

The national instance tests scale. Learning is trained on twelve 256-cell regions, with four development regions. An initial reserved cohort compares architectures; its complete results are in Appendix F. The main learning test below uses sixteen further closed regional instances, with no shipments across their outer boundaries. These smaller problems permit repeated all-route optimization. They use the same US data; the national study is adaptive, while the regional initialization test freezes weights and settings before testing.

Table 1: Roles of the national instance and regional cohorts. Each regional problem contains 256 grid cells. The two test cohorts remain separate; the external learned-method comparison uses the final cohort retrospectively.

cohort	instances	purpose
National	1	Full-scale plan and upper bound
Training	12	Fit material offsets and neural controls
Development	4	Development diagnostics
First reserved test	16	Architecture/control study (appendix)
Final reserved test	16	Frozen learned-versus-zero initialization

We report annual profit, valid upper bounds and gap $100(U - L)/|U|$. Planning gain over a positive baseline X is $100(L - X)/X$. Regional reductions are in \$/year per region; the prespecified uncertainty analysis divides each region's reduction by its annual manure supply and bootstraps regions. Bounds improve by decreasing; these reductions are not additional achieved profit. Our Gurobi 13.0.3 block solves use four CPU threads; the national LP uses 64. PyTorch 2.6 trains the predictors. Full hardware, solver budgets, features and numerical tolerances are in Appendix E. Regional blocks have a 120-second limit and a 10^{-6} relative MIP gap target; national bound blocks have a 300-second limit. National tables use the common numerical replay with rescaled flows and objectives and 10^{-9} primal, dual and integrality tolerances. A separate physical implementation checks returned plans. All per-block statuses, residuals and sensitivity checks are retained in Appendix G.

4.1 National planning and optimality gap

The published national baseline is gSC (Ma & Zavala, 2022), using its original route sampling and balance coarsening on the same LCO formulation. It receives none of our planner, prices or spatial relaxation. Table 2 compares the best returned plans and valid upper bounds under the recorded settings; end-to-end compute budgets differ. Full runs are in Appendix C.

Table 2: National results on the same LCO, in \$B/year. Gain is relative to the best tested gSC plan. Its upper-bound runs reach their limits before completing the root LP.

method	profit	upper bound	gap (%)	gain (%)
gSC, best returned	124.873	763.164	83.64	–
Ours	129.870	132.154	1.73	+4.00

Our feasible \$129.870B/year plan exceeds the best tested gSC plan by 4.00% (\$4.996B/year). Both plans pass independent physical checks. The \$132.154B/year upper bound leaves a 1.73% gap to the dense optimum. All 2,048 blocks contribute finite solver upper bounds, with 2046 reaching optimal status in the common numerical replay.

The stronger LP-price input comes from five column-generation rounds after an initial 32-neighbor LP, adding profitable omitted routes (Lübbecke & Desrosiers, 2005). It takes 101 minutes on 64 threads and reaches 23.1 million variables. This LP has not converged to the full relaxation: its duals supply prices, while the regional integer solves provide the full-model bound. Let $z_R =$ \$135.110B/year be its feasible relaxed objective and z_{LP} the unknown full LP optimum. Extending its flows by zeros gives

$$\mathrm{OPT} \le U = \$132.154\mathrm{B/year} < z_R \le z_{\mathrm{LP}}. \tag{9}$$

Thus the integer-block bound is tighter than the full continuous relaxation without requiring that full relaxation to be solved.

Table 3: Internal price-learning ablation on the same national spatial relaxation. Upper bounds are in \$B/year; reduction is relative to the LP-price control. All rows use the same CG input and material floor.

price field	upper bound	reduction (%)	gap (%)
LP prices, no learning	133.110	0.00	2.43
GNN, seed 0	132.353	0.57	1.88
Cellwise MLP, seed 0	132.413	0.52	1.92
Eight learned offsets	132.154	0.72	1.73

Table 3 isolates price choice. Learned offsets reduce the matched LP-price bound by \$955.5M/year to \$132.154B/year. GNN and MLP controls test feature-dependent corrections on the same relaxation. The floor changes one entry of the offset field and leaves the LP control unchanged; all raw/floored fields and both LP inputs appear in Appendix G. This national comparison is adaptive and does not establish universal network or solver superiority. The plan also reduces modeled emissions by 17% and phosphorus excess by 59% relative to local untreated application; the conditional environmental accounts are in Table 24.

4.2 Learning after further multiplier optimization

An initially tighter bound need not remain better after classical optimization. We therefore compare the same eight-dimensional optimizer from two starts: zero offsets, which retain the LP base, and learned offsets. An earlier exploratory follow-up motivated this test. We then froze the weights and optimizer settings before selecting sixteen further 256-cell regions at least 100 km from earlier regional footprints. Every prediction was recorded before its block solves. These are new regional optimization contexts; their cells had appeared in earlier national work. Appendix F.3 gives the full selection and frozen protocol.

Each region has eight blocks. Both starts receive twelve rounds, or 96 block solves, and all 3,072 solves reach optimal status. The learned start gives tighter bounds in 15/16 regions after twelve rounds, by a mean \$0.570M/year per region beyond the zero start. Before refinement, the learned field improves on its LP-price control in all sixteen by \$2.981M/year per region. Figure 2 shows how much of this initial advantage remains as both starts improve. Each start constructs its own block models and then reuses them within that run. The reported bound is the smallest valid upper bound returned within the round budget. Thus both starts receive the same number of block solves and the same refinement rules; the learned parameters change the initial point. Training and development outcomes do not enter the final test's regional averages.

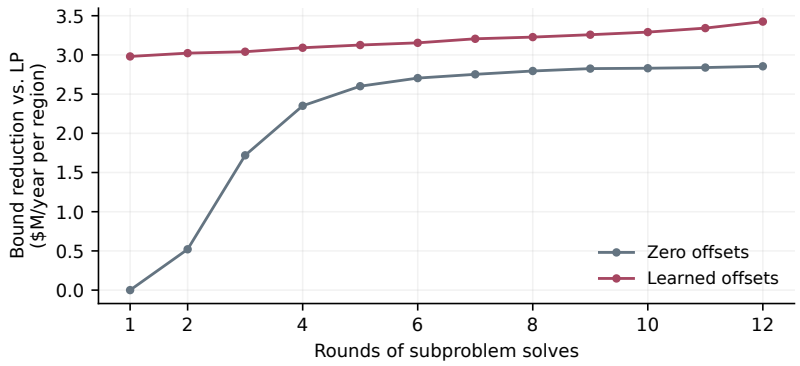


Figure 2: Mean annual bound reduction per region from LP prices on the final sixteen-region test. Both starts use the same optimizer and twelve rounds, including the initial evaluation. Weights and refinement settings were fixed before testing.

The prespecified supply-normalized advantage is \$0.094/t, with a 95% paired regional-bootstrap interval of $[0.056, 0.139]$ from 20,000 resamples. This interval describes the normalized statistic, not the annual-dollar mean. A numerical replay preserves all win/loss signs and the rounded statistic. At twelve rounds, median online times are 45.8 seconds from learned offsets and 47.1 from zero, excluding common 6.1-second setup and 51.3-minute offline offset training. These timings are descriptive. The frozen comparison supports reuse of the learned initialization within this model family.

4.3 Published algorithms on the same LCO

The learned baseline is the MILP-graph multiplier predictor of Demelas et al. (2024). Its separate LCO interface duplicates each shipment and relaxes agreement between sending and receiving copies; it receives none of our spatial bound, price processing or learned offsets. We reproduce its eight LP/MILP features, five residual graph blocks, Gaussian decoder and bound-driven objective with one seed and fixed settings. Training uses the same twelve regions and 360 updates; evaluation uses the final sixteen regions and the published best-of-five stochastic prediction rule. All eighty bound evaluations reach optimal status. Appendix D.2 documents the implementation.

Our learned bounds are lower in every region, by a mean \$9.018M/year. Our LP-only spatial relaxation already improves on the external method by \$6.038M/year (Table 4); the additional learned

Table 4: Complete-method comparison on the final sixteen regions. Extra tightening is relative to the Demelas implementation's best-of-five bound. Our rows use one round before refinement. Relaxations, LP inputs and evaluation costs differ.

bound method	extra tightening ($M/year)	tighter regions
Demelas et al., best of five	reference	–
Ours, LP prices	6.038	16/16
Ours, eight learned offsets	9.018	16/16

calibration contributes $2.981M/year. This is a complete-method comparison, while the matched two-start test isolates learning. The external comparison is retrospective on the already evaluated final cohort, with every case retained. The external predictor improves its own LP initialization in all sixteen regions, by a mean $0.017M/year.

We also implement the callback Benders algorithm of Chen et al. (2026) on the sixteen original regional LCOs (twelve training and four development cases). At the 1,800-second limit its median gap is 15.15%, versus 0.89% for retained direct-Gurobi runs; all returned plans pass physical checks. Solver versions, scaling and hardware differ, so this is a recorded solution-quality comparison. Complete results and adaptations are in Appendices D.1 and C.

4.4 Model capacity and computational cost

Table 5 compares internal predictors on the first reserved cohort, separate from the final initialization test. All share the LP base, spatial relaxation and training regions, with one round per field. Eight offsets give the largest mean reduction among these seed-0 fields; Appendix F reports all seeds.

Table 5: Internal architecture ablation on the first sixteen test regions (seed 0). Reduction is relative to matched LP prices, in $M/year per region; higher is better. Wins count improvements over LP prices.

correction	learned parameters	mean reduction	wins
Physical-cell GNN	93,576	4.425	15/16
Cellwise MLP	6,024	4.965	14/16
Eight material offsets	8	4.997	16/16

National planning uses 64 parallel block MIPs with 40-minute caps, followed by a national flow LP taking about 12 minutes on 64 threads. Bound computation has a separate cost: the CG input takes 101 minutes, and the floored offset field takes 5,182 seconds summed across its 2,048 four-thread block solves. This sum excludes input construction and preprocessing and is not elapsed cluster time; blocks can run in parallel. The frozen offsets require no national retraining. Appendix G reports all field timings and budgets.

5 Conclusion

In this paper, we have proposed a framework for national-scale manure life-cycle optimization, an important problem for recovering agricultural nutrients, coordinating processing infrastructure and managing the environmental burden of livestock production. The central challenge is to make national planning tractable while assessing solution quality across billions of possible transport routes. Our spatial decomposition addresses this challenge by preserving local material balances, transport and integer facility decisions while bounding the value of all cross-region shipments. We further introduce eight shared, learned adjustments to LP-derived prices, retaining geographic information while enabling transfer across regions and convex refinement. Together, these contributions connect feasible sparse-network plans to upper bounds on the full national optimum. The national results establish a 1.73% optimality gap, while controlled regional experiments demonstrate the benefit of learned initialization. This work shows how physical problem structure and compact multiplier learning can support large-scale agricultural planning with explicit guarantees on solution quality.

AI USE STATEMENT

Large language models (LLMs) were used to assist with manuscript editing, grammar checking, and code implementation. The authors take full responsibility for all scientific claims, code, data rights, and the final submission.

REPRODUCIBILITY STATEMENT

We will provide our code, regional datasets, trained checkpoints and baseline implementations, together with saved results and scripts for checking feasibility and reproducing the reported numbers. The full formulation and parameter values are given in Appendix A. Experimental settings are described in Appendix E, and proofs are provided in Appendix I. Running the national experiments also requires the road-distance cache described in Appendix E.

REFERENCES

Ahmed Abbas and Paul Swoboda. DOGE-Train: Discrete optimization on GPU with end-to-end training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 20623–20631, 2024.

Daniel Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

Usman Akram, Nils-Hassan Quttineh, Uno Wennergren, Karin Tonderski, and Geneviève S. Metson. Optimizing nutrient recycling from excreta in sweden and pakistan: Higher spatial resolution makes transportation more attractive. *Frontiers in Sustainable Food Systems*, 3:50, 2019. doi: 10.3389/fsufs.2019.00050.

Swann Bessa, Darius Dabert, Max Bourgeat, Louis-Martin Rousseau, and Quentin Cappart. Learning valid dual bounds in constraint programming: Boosted lagrangian decomposition with self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11113–11121, 2025.

Yanju Chen, Yan Liu, and Yuhan Wang. Designing an agricultural waste collection and transport network based on robust optimization. *Socio-Economic Planning Sciences*, 103:102375, 2026. doi: 10.1016/j.seps.2025.102375.

E. Cornielje, B. Markhorst, A. Zocca, and R. van der Mei. LOTUS: A warm-start framework for powering dual decomposition in large-scale two-stage stochastic programs, 2026. arXiv:2603.01733.

F. Demelas, J. Le Roux, M. Lacroix, and A. Parmentier. Predicting lagrangian multipliers for mixed integer linear programs. In *Proceedings of the International Conference on Machine Learning*, pp. 10368–10384, 2024.

F. Demelas, J. Le Roux, A. Frangioni, M. Lacroix, E. Traversi, and R. Wolfler Calvo. Bundle network: A machine-learning-based bundle method, 2025. arXiv:2509.24736.

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Stéphane Eilles-Chan Way, Hugo Percot, Quentin Cappart, Tias Guns, and Louis-Martin Rousseau. Scaling decision-focused learning to large problems with lagrangian decomposition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2217–2226, 2026.

FAO. Gridded livestock density (global – 2020 – 10 km) – GLW4. FAO Agro-Informatics Data Catalog, 2024. Five-arcminute cattle density dataset.

Marshall L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

Antonio Frangioni. Standard bundle methods: Untrusted models and duality. In *Numerical Nonsmooth Optimization*, pp. 61–116. 2020.

Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact combinatorial optimization with graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Arthur M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

Mohammad T. Hajiaghayi, Mohammad Mahdian, and Vahab S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1):42–47, 2003.

Q. Han, L. Yang, Q. Chen, X. Zhou, D. Zhang, A. Wang, R. Sun, and X. Luo. A GNN-guided predict-and-search framework for mixed-integer linear programming. In *International Conference on Learning Representations*, 2023.

Yicheng Hu, Apoorva M. Sampat, Gerardo J. Ruiz-Mercado, and Victor M. Zavala. Logistics network management of livestock waste for spatiotemporal control of nutrient pollution in water bodies. *ACS Sustainable Chemistry & Engineering*, 7(22):18359–18374, 2019.

International Food Policy Research Institute. Global spatially-disaggregated crop production statistics data for 2020. Harvard Dataverse, 2024. SPAM 2020 dataset.

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*, pp. 10219–10227, 2022.

Tung Quoc Le, Anh Tuan Nguyen, and Viet Anh Nguyen. Provably data-driven lagrangian relaxation for mixed integer linear programming. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2605.19052`.

Claude Lemaréchal. An extension of davidon methods to non differentiable problems. *Mathematical Programming Study*, 3:95–109, 1975.

Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005. doi: 10.1287/opre.1050.0234.

Jiaze Ma and Victor M. Zavala. Solution of large-scale supply chain models using graph sampling & coarsening. *Computers & Chemical Engineering*, 163:107832, 2022. doi: 10.1016/j.compchemeng.2022.107832.

Vinod Nair et al. Solving mixed integer programs using neural networks, 2020. arXiv:2012.13349.

Seonho Park and Pascal Van Hentenryck. Self-supervised primal-dual learning for constrained optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4052–4060, 2023.

Nathan Preuss and Fengqi You. Consequential versus attributional life cycle optimization of poultry manure management technologies in a food-energy-water-waste nexus. *Journal of Cleaner Production*, 469:143133, 2024. doi: 10.1016/j.jclepro.2024.143133.

Apoorva M. Sampat, Yicheng Hu, Mahmoud Sharara, Horacio Aguirre-Villegas, Gerardo Ruiz-Mercado, Rebecca A. Larson, and Victor M. Zavala. Coordinated management of organic waste and derived products. *Computers & Chemical Engineering*, 128:352–363, 2019.

Mahmoud Sharara, Apoorva Sampat, Laura W. Good, Amanda S. Smith, Pamela Porter, Victor M. Zavala, Rebecca Larson, and Troy Runge. Spatially explicit methodology for coordinated manure management in shared watersheds. *Journal of Environmental Management*, 192:48–56, 2017.

Richard M. Soland. Optimal facility location with concave costs. *Operations Research*, 22(2): 373–382, 1974.

Yinghua Yin, Shufang Guo, Qiyu Xu, Jian Liu, Hongyuan Wang, Yanhua Zhuang, Zhen Wang, Lingling Hua, Hongbin Liu, and Limei Zhai. Co-benefits for cropland yield, nitrogen emissions, and climate impact through multi-objective optimization agricultural manure solutions. *Nature Communications*, 16:6415, 2025. doi: 10.1038/s41467-025-61885-w.

Paul H. Zipkin. Bounds on the effect of aggregating variables in linear programs. *Operations Research*, 28(2):403–418, 1980a.

Paul H. Zipkin. Bounds for row-aggregation in linear programming. *Operations Research*, 28(4): 903–916, 1980b.

A COMPLETE MODEL FORMULATION

This appendix states the instance family used in the paper, in the form implemented in `gmodel.py` and solved in every experiment of Section 4. We use the supplied reference life-cycle optimization formulation and material inventory, with the changes in Section A.9 agreed with its developer. The physical formulation is a study input, not a methodological contribution. The main text (Section 3) suppresses the per-cell agronomic variables and the accounting; both are restored here. Throughout, one freight mode (truck) exists, so the mode index is dropped.

A.1 SETS AND INDICES

symbol	index	meaning	cardinality
$\mathcal{M}$	m	materials	8
$\mathcal{C}$	l, A, B	grid cells (10 km, Equal Earth)	63,873 (national)
$\mathcal{O}$	o	processing operations	3 (AD, SLS, Py)
$\mathcal{P}$	p	crops	29
$\mathcal{N}$	n	nutrients {N, P, K}	3

The eight materials and their inventory identifiers are raw manure (`inf`), separated manure solids (`infS`), separated manure liquid (`infL`), biochar from manure solids (`infSPy`), digestate (`dig`), separated digestate solids (`digS`), separated digestate liquid (`digL`) and biochar from digestate solids (`digSPy`). The notation `raw` in this formulation refers to `inf`. The subset $\mathcal{M}^{\mathrm{Py}} \subset \mathcal{M}$ contains the two biochars, which retain their distinct inventory properties. Each operation consumes one input material: anaerobic digestion (AD) turns raw manure into digestate, solid–liquid separation (SLS) splits raw manure or digestate into solids and liquids, and pyrolysis (Py) turns either solids fraction into biochar.

A.2 DATA

symbol	meaning	units
b_l	manure supply at cell l: herd size $\times\ \mu\, \varrho$	t/yr
d_{AB}	road distance between cells	km
$\alpha_{m'mo}$	yield of m from one tonne of m' under operation o	t/t
β_m	storage retention fraction of m	–
γ_{mn}	mass fraction of nutrient n in material m	t/t
$K_o^{\max},\ K_o^{\mathrm{mid}}$	capacity ceiling and cost breakpoint of operation o	t/yr
$V_o^0,\ V_o^1$	capital cost at K_o^{mid} and $K_o^{\max}$	\$
ρ_o	capital annuity factor, $\rho_o = r_o/\big(1-(1+r_o)^{-n_o}\big)$	1/yr
$\phi_m,\ \nu_m$	fixed and per-km freight rate	\$/t, \$/(t km)
$c_m^{\mathrm{app}},\ c_o^{\mathrm{om}},\ c_m^{\mathrm{sto}}$	application, operating, storage cost	\$/t
$r_m^{\mathrm{mat}},\ r_p^{\mathrm{crop}}$	material and crop sale prices	\$/t
$H_{lp},\ Q_{lp}^{\max}$	harvested area and attainable production	km^2, t/yr
R_{lpn}	nutrient recommendation, $R_{lpn} = H_{lp} \cdot \bar{R}_{pn}$	t/yr
R_{lp}^{char}	biochar recommendation, $H_{lp} \cdot \bar{R}_p^{\mathrm{char}}$	t/yr
$Y_p^{\min},\ Y_p^{\mathrm{add}}$	baseline yield fraction; biochar yield bonus	–
$e^{(\cdot)}$	emission factors (reported, not priced; see §A.7)	t CO_2e/t, /\$

A.3 DECISION VARIABLES

Continuous, all nonnegative: transport x_{mAB} (t/yr); processing consumption C_{mlo} (production C_{mlo}^{out} is derived in Eq.(11)); material routed through storage s_{ml}; application to a modelled crop row a_{mlp}, and application outside the modelled rows a_{ml}^0 (§A.6); biochar yield bonus $w_{lp} \in [0, Y_p^{\mathrm{add}}]$ and actual yield fraction u_{lp}; segment weights $\lambda_{lo}^0, \lambda_{lo}^1, \lambda_{lo}^2 \in [0,1]$. Binary: segment decisions $z_{lo} \in \{0,1\}$ for every operation with a cost breakpoint. There are no other integer variables. Capacity $\kappa_{lo} = \sum_m C_{mlo}$ and capital cost χ_{lo} are affine in the variables above and are written out for readability.

A.4 MATERIAL FLOW

Material balances connect the planning decisions: processing changes material type, transport changes location, and storage leads to local application. Transport sums below run over allowed directed routes between distinct cells.

(M1) Mass balance. At each material–cell pair, storage, exports and processing consumption equal supply, imports and processing output; exogenous supply enters only for raw manure:

$$s_{ml} + \sum_{B} x_{mlB} + \sum_{o} C_{mlo} = b_l \, \mathbf{1}[m = \texttt{raw}] + \sum_{A} x_{mAl} + \sum_{o} C^{\text{out}}_{mlo} \qquad \forall m, l. \tag{10}$$

(M2) Conversion. Production is the yield table applied to consumption; a material that is not an input of o has no consumption variable for o:

$$C^{\text{out}}_{mlo} = \sum_{m'} \alpha_{m'mo} \, C_{m'lo} \qquad \forall m, l, o. \tag{11}$$

(M3) Storage to application. Material surviving storage losses is applied in the cell where it is stored, either to a modelled crop row or outside one:

$$\beta_m \, s_{ml} = \sum_{p} a_{mlp} + a^0_{ml} \qquad \forall m, l. \tag{12}$$

The original model's constraint (M4), $\sum_o C_{\texttt{raw},l,o} \le b_l$, which restricted raw-manure processing at a cell to that cell's own supply, is *not* imposed; see Section A.9.

A.5 FACILITIES AND CAPITAL COST

For every cell and operation the capital cost is a concave piecewise-linear function of capacity through the points $(0,0)$, $(K^{\text{mid}}_o, V^0_o)$ and $(K^{\max}_o, V^1_o)$. With the reference parameters the marginal cost falls at the breakpoint $K^{\text{mid}}_o = 20{,}000$ t/yr from \$71.3 to \$8.8 per tonne for digestion and from \$400 to \$140 per tonne for pyrolysis; separation has $K^{\text{mid}} = 0$ and is linear at \$3.68 per tonne. Allowing unrestricted interpolation between the endpoints would understate these concave costs. We therefore use one binary per facility with a cost breakpoint to select the active segment and restrict the convex-combination weights to that segment:

$$\kappa_{lo} = \sum_{m} C_{mlo} \qquad \forall l, o, \tag{13}$$

$$\kappa_{lo} = K^{\text{mid}}_o \, \lambda^1_{lo} + K^{\max}_o \, \lambda^2_{lo}, \qquad \chi_{lo} = V^0_o \, \lambda^1_{lo} + V^1_o \, \lambda^2_{lo} \quad \forall l, \; o : K^{\text{mid}}_o > 0, \tag{14}$$

$$\lambda^0_{lo} + \lambda^1_{lo} + \lambda^2_{lo} = 1, \qquad \lambda^0_{lo} \le 1 - z_{lo}, \qquad \lambda^2_{lo} \le z_{lo} \qquad \forall l, \; o : K^{\text{mid}}_o > 0, \tag{15}$$

$$\chi_{lo} = \left(V^1_o / K^{\max}_o\right) \kappa_{lo}, \qquad \kappa_{lo} \le K^{\max}_o \qquad \forall l, \; o : K^{\text{mid}}_o = 0. \tag{16}$$

When $z_{lo} = 0$ the weight λ^2 vanishes and (κ, χ) lies on the first segment; when $z_{lo} = 1$ the weight λ^0 vanishes and it lies on the second. The ceiling $\kappa_{lo} \le K^{\max}_o$ is implied. Every reported physical node remains a single cell; the spatial partitions do not aggregate capital-cost curves or crop decisions. The build binary y_{lo} of the original model, whose only role was the floor $10^{-3} y_{lo} \le \kappa_{lo} \le K^{\max}_o y_{lo}$, carries no objective coefficient and is dropped; no minimum capacity is enforced, and the segment binary z_{lo} is the whole of the integer structure.

A.6 AGRONOMY

Crop application affects profit through the yield response. The model allows nutrient shortfalls and represents their effect through upper limits on yield, together with a separate biochar bonus. A crop row (l, p) is *modelled* if $H_{lp} \ge 10^{-4}\,\text{km}^2$ and $Q^{\max}_{lp} > 0$; the set of modelled rows is $\mathcal{A}$. Rows below that area contribute their baseline revenue $r^{\text{crop}}_p Q^{\max}_{lp} Y^{\min}_p$ as a constant and receive no application. This threshold avoids very large reciprocal-recommendation coefficients on tiny crop areas. For $(l, p) \in \mathcal{A}$, with the Liebig slope $\sigma_{lpn} = (1 - Y^{\min}_p)/R_{lpn}$ when $R_{lpn} > 0$ and $\sigma_{lpn} = 0$ otherwise:

(P1) Liebig yield line, one per nutrient, and **(P2) unit cap**, both relaxed by the biochar bonus:

$$u_{lp} \le Y_p^{\min} + w_{lp} + \sigma_{lpn} \sum_m \gamma_{mn}\, a_{mlp} \qquad \forall (l,p) \in \mathcal{A},\ n, \tag{17}$$

$$u_{lp} \le 1 + w_{lp} \qquad \forall (l,p) \in \mathcal{A}. \tag{18}$$

(P3) Biochar bonus. The source model's bonus is the saturating piecewise-linear interpolant through $(0,0)$, $(R_{lp}^{\text{char}}, Y_p^{\text{add}})$ and $(10R_{lp}^{\text{char}}, Y_p^{\text{add}})$ of the biochar applied. It is concave and the objective rewards w_{lp}, so its hypograph is exact:

$$w_{lp} \le \frac{Y_p^{\text{add}}}{R_{lp}^{\text{char}}} \sum_{m \in \mathcal{M}^{\text{Py}}} a_{mlp}, \quad w_{lp} \le Y_p^{\text{add}}, \quad \sum_{m \in \mathcal{M}^{\text{Py}}} a_{mlp} \le 10\, R_{lp}^{\text{char}} \quad \forall (l,p) \in \mathcal{A} : R_{lp}^{\text{char}} > 0, \tag{19}$$

and $w_{lp} = 0$ where $R_{lp}^{\text{char}} = 0$. Crop output is $q_{lp} = u_{lp}\, Q_{lp}^{\max}$.

No nutrient-coverage requirement. The original model contains a row $\xi_{ln} = \sum_p g_{lpn} - \sum_p R_{lpn}$ with $\xi_{ln} \ge 0$ by variable domain, which requires the aggregate nutrient application at every cell to meet the aggregate recommendation. That requirement is feasible for New York, which has a manure surplus, and infeasible nationally: manure supplies about 45% of nitrogen, 35% of phosphorus and 73% of potassium demand. The corrected model imposes *no* coverage requirement. A shortfall lowers yield through Eq.(17) and nothing else; an excess costs nothing beyond storage and application. Excess nutrients are reported after the solve as the positive part $\xi_{ln} = \max\{0, \sum_p \sum_m \gamma_{mn} a_{mlp} - \sum_p R_{lpn}\}$, which is what the original row computes whenever the requirement is met.

A.7 Accounting and objective

Revenue comes from crop output and the reference value assigned to stored materials. The annual objective subtracts capital annuities, application, operation, freight and storage costs:

$$R^{\text{mat}} = \sum_{m,l} r_m^{\text{mat}}\, s_{ml}, \quad R^{\text{crop}} = \sum_{l,p} r_p^{\text{crop}}\, q_{lp}, \qquad R = R^{\text{mat}} + R^{\text{crop}}, \tag{20}$$

$$T^{\text{cap}} = \sum_{l,o} \rho_o\, \chi_{lo}, \qquad T^{\text{app}} = \sum_{m,l} c_m^{\text{app}} \Big(\sum_p a_{mlp} + a_{ml}^0 \Big), \quad T^{\text{om}} = \sum_{m,l,o} c_o^{\text{om}}\, C_{mlo}, \tag{21}$$

$$T^{\text{fr}} = \sum_{m,A,B} \big(\phi_m + \nu_m\, d_{AB} \big)\, x_{mAB}, \qquad T^{\text{sto}} = \sum_{m,l} c_m^{\text{sto}}\, s_{ml}. \tag{22}$$

The single objective is net profit,

$$\max\ R - T^{\text{cap}} - T^{\text{app}} - T^{\text{om}} - T^{\text{fr}} - T^{\text{sto}}. \tag{23}$$

Emissions E (from application, storage, production, freight and capital, with the factors and the five expressions of the source model) and excess nutrients $\Xi_n = \sum_l \xi_{ln}$ are computed from the exported plan and reported in Section 4.1; neither enters the objective. The carbon price and the nutrient penalties of the original model are set to zero. The emission-minimizing, lexicographic and Pareto-sweep objectives of the source study are not used.

A.8 Price-field transformation and road distances

Whether one price-transformation pass reaches a fixed point depends on the road costs. We therefore check the supplied distances before using properties associated with metric networks. The supplied matrix d is used as given, with $c_{mAB} = \phi_m + \nu_m d_{AB}$ for $A \ne B$ and $c_{mAA} = 0$. It does not obey a triangle inequality. In a seeded sample of 256 national cells, 22,744 of 65,536 ordered pairs have a direct distance more than 1 km longer than the shortest two-leg distance through another sampled cell; the maximum excess is 398 km. For raw manure freight, 17,208 sampled ordered pairs have a cheaper two-leg shipment even after paying the fixed charge twice. Thus the mathematical idempotence and arbitrage-free properties of a metric min-plus closure cannot be invoked for this

instance. The implemented T in Eq.(4) uses one pass per regional call. The national 32-neighbor-input study uses three passes per call; the CG-input study iterates to a checked fixed point, with a 64-pass ceiling (Appendix G). Sparse inputs are transformed before the network, and its residual output is transformed again as specified in Algorithm 1; national output also receives a per-material floor. These transformations may change bound quality, but every resulting field remains admissible for the Lagrangian inequality in Eq.(3).

A.9 Changes relative to the source model

1. **Coverage requirement removed** (§A.6): the per-cell requirement that applied nutrients meet the recommendation is dropped, because it is infeasible for the national instance. Excess is reported as a positive part.
2. **(M4) dropped**: raw manure may be processed at any cell, not only where it was produced. The author confirmed the original row was a guard against a solver artefact rather than a physical restriction.
3. **No policy cost**: carbon price and nutrient penalties are zero; emissions are reported.
4. **Segment binaries** (§A.5) replace the build binary and the SOS2 piecewise-linear constraints of the source implementation; the cost function is unchanged. The 10^{-3} t floor is not enforced.
5. **Biochar bonus** as the hypograph inequalities Eq.(19) instead of an SOS2 piecewise-linear constraint; exact because the bonus is concave and rewarded.
6. **Crop rows below** 10^{-4} **km**^{2} are carried as constants.

A.10 Reference parameters (US-ref-v1)

parameter	value	source / status
manure per head μ	14.4 t/yr	supplied reference value; provisional
recoverable fraction ϱ	1.0	scenario axis
material properties β_m, γ_{mn}, emission factors	reference material inventory	full precision, closed element balances
pyrolysis yields α	`infS`→`infSPy` 0.093359; `digS`→`digSPy` 0.068390	author's SI inventory
other yields α	AD 0.96036; SLS 0.29 solids, 0.71 liquids	author's SI inventory
$K_o^{\max}$ (single plant)	AD 453,600; SLS 652,800; Py 320,000 t/yr	author's US inventory
K_o^{mid}; V_o^0, V_o^1	AD: 20,000; \$1.426M, \$5.244M. Py: 20,000; \$8M, \$50M. SLS: 0; –, \$2.4M	author's US inventory
biochar subsidy $r_{\mathrm{Py}}^{\mathrm{mat}}$	\$200/t	scenario axis (0–400 in the source study)
biochar yield bonus Y_p^{add}	0.10 (all crops)	scenario axis (0–0.10 in the source study)
baseline yield $Y_p^{\min}$	0.74 (all crops)	source study
minimum modelled crop area	10^{-4} km^2	numerical

A.11 Implementation notes

Solves. Block and regional mixed-integer programs are solved with Gurobi at a relative MIP gap of 10^{-4} to 10^{-6}; the Lagrangian block subproblems of Section 3.2 use 10^{-6} and a solver dual bound is recorded whenever a time limit is hit, since any valid bound on a block preserves the validity of Eq.(3). The national fixed-segment linear program is solved by barrier with crossover (convergence tolerance 10^{-9}), which yields a vertex solution that the checker accepts; pricing LPs used only to seed price fields run barrier without crossover. The external baseline settings are specified separately in Appendices C and D.

Independent verification. The exported national hub plan is revalidated by `verify_corrected.py`, a standalone checker that re-derives Eq.(10)–Eq.(22), the yield chain and the capital-cost curve from the raw instance files with array arithmetic only (no solver, no modelling library). Each physical constraint is tested row by row at a relative tolerance of 10^{-4} (with scale floored at one tonne, dollar or yield unit); accounting totals are recomputed from the exported physical tables and compared to the reported ones. For the national plan the worst relative violation is 6×10^{-8} and the exported objective equals the solver's to 10^{-7} relative. The internal nearest-neighbor planner ablation was separately reconstructed from its frozen block decisions, solved, exported and checked by the same procedure; its maximum relative violation is also 5.96×10^{-8}. That internal comparison retains optimizer objectives consistently. Independently recomputed export accounts differ from them by less than \$830/year for each plan, below 7×10^{-9} relative. For the external gSC comparison, a separate array-based checker reads the original solver decisions without rounding, repair or reoptimization. It checks cell balances, storage losses, application, crop responses, capital segments, route costs and total profit. All three returned gSC plans pass; the best plan's recomputed profit differs by less than \$0.001/year. Appendix C reports the full runs.

Correspondence with earlier implementations. The model builder (`gmodel.py`) is a fresh implementation. Under the legacy settings of the previous audited implementation (M4 kept, coverage requirement, original inventory) it reproduces that implementation's optimal values to 5×10^{-9} relative on dense 24-cell instances.

A.12 SYMBOL–IMPLEMENTATION CORRESPONDENCE

symbol	`gmodel.py`	source data
x_{mAB}	`x[k,i]` over the arc list	–
C_{mlo}	`u[m0,o,a]`	–
s_{ml}	`S[k,a]`	–
a_{mlp}, a^0_{ml}	`w[k,l,p]`, `w0[k,l]`	–
u_{lp}, w_{lp}	`Y[l,p]`, `eta[l,p]`	–
$\lambda^{0,1,2}_{lo}$, z_{lo}	`l0`, `l1`, `l2`, `seg[o,a]`	–
b_l	`inst['supply']`	`Location.cattle` × `REF.manure_t_per_head` × `REF.recoverable`
α, β_m, γ_{mn}	`alpha`, `beta`, `phi`	`Mat0_Mat1_Operation.Yield`, `Material.StoLoss`, `Material.N/P/K` (SI inventory)
$K^{\max}_o$, K^{mid}_o, V^0_o, V^1_o	`mx`, `mid`, `v0`, `v1`	`Operation` sheet
ρ_o	`annuity(r,T)`	`Operation.InterestRate`, `FacilityLifetime`
ϕ_m, ν_m	`fr[k]['f0']`, `fr[k]['f1']`	`Material_Freight` sheet
H_{lp}, $Q^{\max}_{lp}$	`area`, `prod`	`Location.`p`_harvested`, p`_production`
R_{lpn}, R^{char}_{lp}	`area*rec`, `B`	`Product.N/P/K/Char` × area
$Y^{\min}_p$, Y^{add}_p	`ymin`, `dl`	`Product` sheet; `REF.biochar_bonus_scale`
$r^{\mathrm{mat}}_{\mathrm{Py}}$	`Material.Revenue`	`REF.biochar_subsidy`

B PHYSICAL NOTATION AND IMPLEMENTATION DETAILS

This appendix gives the physical-variable definitions, price transformations, training settings and complete refinement procedure summarized in the main text. It preserves the formulation and fixed settings of the reported experiments. Here $l, A, B \in \mathcal{C}$ index cells, $m \in \mathcal{M}$ materials, $o \in \mathcal{O}$ processing technologies and $p \in \mathcal{P}$ crops.

B.1 PHYSICAL QUANTITIES AND CAPITAL SEGMENTS

Table 6: Physical decisions and derived quantities. All quantities are nonnegative; C^{out}_{mlo}, κ_{lo} and χ_{lo} are affine expressions in the decision variables. Flows and capacities are annual.

symbol	meaning	units
x_{mAB}	Material m shipped from cell A to cell B, $A \neq B$	t/yr
C_{mlo}, C^{out}_{mlo}	Material consumed and produced by technology o at cell l	t/yr
s_{ml}	Material entering storage at cell l	t/yr
a_{mlp}, a^0_{ml}	Application to crop p and outside modeled crop rows	t/yr
u_{lp}, w_{lp}	Crop yield relative to attainable production and its biochar bonus	–
$\kappa_{lo} = \sum_m C_{mlo}$	Processing capacity at cell l for technology o	t/yr
χ_{lo}	Plant capital cost, before annualization	$
$\lambda^j_{lo} \in [0,1]$	Interpolation weights at the three cost-curve points, $j = 0, 1, 2$	–

Facility capacity and capital cost. Digestion and pyrolysis have two-segment, concave piecewise-linear capital-cost curves: the marginal cost of adding capacity falls above a breakpoint. Write K^{mid}_o for this breakpoint and $K^{\max}_o$ for the maximum annual capacity. At each cell l, the binary z_{lo} chooses which segment applies:

$$\begin{aligned} z_{lo} = 0: &\quad 0 \le \kappa_{lo} \le K^{\text{mid}}_o, \\ z_{lo} = 1: &\quad K^{\text{mid}}_o \le \kappa_{lo} \le K^{\max}_o, \end{aligned} \qquad l \in \mathcal{C}, \quad o \in \mathcal{O}_{\text{pw}} = \{\text{AD}, \text{Py}\}. \tag{24}$$

Capacity remains continuous within the selected range. The weights λ^j_{lo} correspond to zero capacity, the breakpoint and maximum capacity for $j = 0, 1, 2$, respectively. Only the selected segment's two endpoints are used to interpolate capacity and capital cost. Restricting interpolation to adjacent endpoints prevents understating the concave cost by using a straight line from zero to maximum capacity. Separation has a linear capital cost and needs no segment binary. Thus the complete binary vector is $z \in \{0,1\}^{n_z}$, where $n_z = n|\mathcal{O}_{\text{pw}}| = 2n$. Appendix A.5 gives the linear constraints linking z, the weights, capacity and capital cost.

B.2 A MATERIAL-PRICE FLOOR

A separate transformation uses local disposal to limit the effect of very low multipliers. The model allows material to be stored and applied locally outside the modeled crop rows. Let v^{loc}_m be the net value in $/t of this option per tonne entering storage, including application after storage losses. This value is a model parameter; s_{ml} in Table 6 is the stored quantity.

Proposition 5 (A dominating material-price floor)**.** *Suppose this local option is available at every cell without a quantity cap and freight costs are nonnegative. For any material-wide $f_m \le v^{\text{loc}}_m$, let $\pi'_{mB} = \max\{\pi_{mB}, f_m\}$. Then $\mathcal{L}_k(\pi') \le \mathcal{L}_k(\pi)$ for every block, so $U(\pi') \le U(\pi)$.*

Any export made more valuable by flooring has netback at most f_m and can instead use the local option. Returning import prices to the original lower field can only improve that block solution. This proves the inequality; Appendix I details the construction and checks the reference inventory's conservative floor. The result concerns exact optima, while time-limited solver bounds retain their actual statuses. Bound validity itself holds for every field by Eq.(3).

B.3 TRAINING AND NEURAL CONTROLS

Each internal model receives 30 epochs on twelve training regions, one update per region per epoch in a seeded random order. Adam uses learning rate 0.05 for offsets and 0.002 for the neural models; parameter-gradient norms are clipped to five. All 360 updates per model have optimal block solutions. The checkpoint is epoch 30 regardless of development scores. All models in this comparison train against the same valid per-cell-cap objective.

To test whether the correction benefits from location-specific features, we compare the shared offsets with a cellwise multilayer perceptron (MLP) and a physical-cell GNN. Each GNN node represents one grid cell; directed edges connect it to its sixteen nearest road neighbors. Their 20 input

channels comprise supply, crop area, N/P/K needs, crop value and 15/30/60-km supply/phosphorus catchments, plus eight LP prices divided by ten. Nonnegative physical features use the transform $\log(1+\cdot)$; all channels then use means and standard deviations fitted on training cells only, with 10^{-6} added to each standard deviation. The GNN has three width-64 layers with mean/max aggregation and edge-distance features; its output head starts at zero. The MLP bypasses these graph layers: it has 6,024 active parameters versus 93,576 for the GNN. Their predicted residual field $\delta \in \mathbb{R}^{8n}$ contains a separate adjustment δ_{mB} for each material and cell, evaluated through $T(q+\delta)$ with the declared pass counts. These alternatives test whether a feature-dependent correction improves on the eight offsets. Appendix J reports the earlier checkpoints and component studies.

B.4 Convex refinement and its complete procedure

One *round* computes an upper bound by solving every block once at fixed multipliers; the solves may run in parallel. Evaluating the learned initialization uses LP prices, the learned adjustments and one round, with no subsequent multiplier updates. Further refinement counts this initial round in its budget. In the 256-cell tests below, each region has eight blocks, so twelve rounds use $12 \times 8 = 96$ block solves. Offline training is reported separately. Algorithm 2 gives the full deployment procedure. Its integer budget $S \geq 1$ counts rounds, and U stores the smallest computed upper bound. The learned offsets are frozen before deployment; optional refinement then updates a copy for the current instance.

Algorithm 2 Planning and upper bounds with material-price calibration

Require: instance; planning and bound partitions; frozen offsets θ; round budget S; LP input
1: Solve each hub-restricted planner MIP; retain feasible capital segments $z^\star$
2: Fix $z = z^\star$ and solve the national sparse flow LP; verify its plan and record L
3: Compute the chosen LP-price input and its processed base field P
4: **if** $S = 1$ **then**
5: $\quad \pi \leftarrow P + \theta$; optionally apply the declared material floor
6: $\quad$ Solve every block at π; record $U = \sum_k U_k(\pi)$ using solver upper bounds
7: **else**
8: $\quad$ Initialize at θ (or zero for the control); set the best upper bound $U \leftarrow +\infty$
9: $\quad$ **for** $s = 1, \ldots, S$ **do**
10: $\quad\quad$ Evaluate every block at $P + \theta$; retain its feasible incumbent and solver upper bound
11: $\quad\quad$ Update the smallest recorded upper bound U and its field π
12: $\quad\quad$ If an incumbent is unavailable, stop refinement and retain U
13: $\quad\quad$ Add the incumbent affine lower cut of Proposition 4
14: $\quad\quad$ If another evaluation remains, update θ with the proximal master problem
15: **return** checked plan, evaluated price field and interval $[L, U]$

Proof of Proposition 4. The offset is common to all export destinations, so $v_{mA}(P+\theta) = v_{mA}(P) + \theta_m$. Each block's feasible set is independent of prices and its objective is affine in θ for any fixed solution. Maximizing these affine functions and summing over blocks proves convexity. Fixing any feasible block solutions gives an affine lower cut. □

Let t index refinement rounds. At the evaluated offsets $\theta^{(t)} \in \mathbb{R}^8$, let a_t be the sum of the feasible block incumbent objective values, in \$/year. Let $h_t \in \mathbb{R}^8$ collect their total exports minus imports for each material, in t/yr. The cut is given by Eq.(8) in the main text.

The lower cut requires feasible incumbent values; replacing a_t with solver upper bounds can invalidate it. We scale cuts by regional manure supply B_{tot} and retain a center $\bar{\theta}_t \in \mathbb{R}^8$, the currently accepted offset vector. The proximal weight $\mu_t > 0$ penalizes moving far from that center. With scalar $\eta \in \mathbb{R}$ representing the lower approximation to the per-tonne bound, the next trial point solves

$$\begin{aligned} \min_{\theta,\eta} \quad & \eta + \frac{\mu_t}{2}\|\theta - \bar{\theta}_t\|_2^2, \\ \text{s.t.} \quad & \eta \geq V_0/B_{\mathrm{tot}}, \quad \theta \in [-200, 200]^8, \\ & \eta \geq \frac{a_\tau + h_\tau^\top(\theta - \theta^{(\tau)})}{B_{\mathrm{tot}}} \quad \text{for all } \tau \leq t. \end{aligned} \tag{25}$$

Here $\tau \in \{1, \ldots, t\}$ indexes stored cuts, and V_0 is the annual value of a feasible plan with no processing or transport. Since $F(\theta) \geq \mathrm{OPT} \geq V_0$, this plan supplies a constant lower cut. The box $[-200, 200]^8$ limits each offset to \$200/t in magnitude. The initial weight is $\max\{\|h_1/B_{\mathrm{tot}}\|_2/2, 10^{-8}\}$. A trial becomes the center when it realizes at least 0.1 of the positive part of the predicted improvement, measured from the center's per-tonne upper bound to the trial master's η. We then halve μ_t; otherwise we double it, clipping subsequent weights to $[10^{-12}, 10^{12}]$. Both starts use these same settings and twelve evaluations. Refinement optimizes the unfloored affine class; the optional national floor is separate. No convexity of $U \circ T$ or a neural predictor is assumed.

C PUBLISHED GSC COMPARISON AND DIRECT-SOLVER REFERENCE

The national comparison needs both an original-model feasible plan and an upper bound that covers all routes. The two arms of gSC supply these objects through different models. We describe them separately so that a sampled-network bound is not confused with a full-model upper bound.

Algorithm and problem identity. We reproduce graph sampling and coarsening (gSC) of Ma & Zavala (2022) on the same national LCO equations, inventory, location table and road matrix used by our pipeline. The published algorithm has two independent arms. Uniformly sampling allowed directed routes and fixing omitted flows to zero gives a feasible original-model plan. For the upper bound, uniformly sampled geographic pivots define groups by nearest-pivot Haversine distance. Summing material balances within each group eliminates internal freight; each intergroup flow uses the minimum original road cost between the two groups. Local processing, crop response and integer capital decisions retain their original cell indices. This is a relaxation even for directed, nonmetric roads.

The baseline receives none of our hubs, price fields, import-cap relaxation, learned offsets, material floor, route unions, warm starts or final flow LP. Its common-problem compiler only expresses the physical model, with flows in kilotonnes and objective in thousand dollars. On four small cases, its dense optimum agrees with the audited formulation to within $\$10^{-8}$; the checks include missing/nonmetric roads and zero supply. Full route sampling and singleton coarsening recover the dense model, and sampled/coarsened values lie on the appropriate sides of its optimum. These checks validate the problem interface, not performance.

Our Python/Gurobi implementation reproduces the published algorithm; it is not an execution of the authors' Julia example. The archived author code references a 1,372-node coordinate file absent from its public data. We neither substitute a different node table nor count that missing input as an optimization failure. The algorithm and common LCO compiler are provided in the review package.

Configurations and computational budget. We run uniform samples of 10,000, 100,000 and 1,000,000 routes, and upper models with 10 and 30 pivots, all with seed 20260924. The two smaller route counts and pivot counts occur in the author's implementation; the million-route setting increases the sampling resolution without changing the algorithm. Each run uses Gurobi 13.0.3, 32 threads and a 1,800-second solver limit. The lower arm uses relative MIP gap 0.003, as in the author's code; the upper arm uses 0.0001. Primal, dual and integrality tolerances are 10^{-9}, with `NumericFocus=1`. The principal runs use an Intel Xeon Gold 6154 node with approximately 1.5 TB RAM, a 1300-GiB allocation and a 1150-GB solver memory cap.

One separately recorded diagnostic repeats the 30-pivot upper model with Gurobi's standard dual-simplex root method, retaining all equations and tolerances. It uses another CPU node with 32 allocated threads, 256 GiB allocated memory and a 224-GB solver cap. Its result is retained in Table 7; it is not substituted for the default configuration or used to claim a hardware-matched timing improvement. The study contains these six completed executions and one seed. A larger planned sweep was stopped for time; its 22 unstarted configurations are not solver failures. No further tuning or reruns enter these results.

Returned results and interpretation. The 10,000-route run returns the best sampled plan, \$124.873B/year. Its sampled-model gap is 0.0272%, below the configured stopping target, after 161.5 solver seconds and 317.9 seconds in total. The 100,000- and million-route runs reach their

Table 7: Every completed national gSC run. Plans and full-model upper bounds are in $B/year. All returned original-model plans pass the physical checker. Total time includes input loading, construction, optimization and solution saving; checking is separate. A coarsened incumbent is not an original plan, and a sampled-model solver bound does not bound the value attainable through omitted routes.

arm	configuration	plan	upper bound	total min	status
Plan	10,000 edges	124.873	–	5.30	Gap criterion
Plan	100,000 edges	22.339	–	36.38	Time limit
Plan	1,000,000 edges	-1270.507	–	35.35	Time limit
Bound	10 pivots	–	767.681	33.17	Time limit
Bound	30 pivots	–	763.164	35.39	Time limit
Bound	30 pivots; dual simplex	–	763.164	31.95	Time limit

time caps before completing the root LP and return poorer feasible incumbents. These values are retained exactly, without replacing them by a repaired or reoptimized plan. The independent profit reconstruction differs from each recorded objective by less than $0.01/year. Observed total times can exceed the nominal solver allowance because of construction, solver stopping granularity and solution saving.

All upper models also stop before the root LP finishes. Their reported values are the actual `ObjBound` values, not intermediate barrier objectives or coarsened incumbents. The best default bound is $763.164B/year. It pairs with the best sampled plan to give an 83.64% gap; this is the gap obtained within these budgets, not the gap attainable by a converged gSC relaxation.

Our $129.870B/year plan exceeds the best tested gSC plan by $4.996B/year (4.00%). This is a solution-quality comparison on an identical physical model under the recorded settings. The end-to-end compute budgets differ, and one seed on one national instance does not establish a general speedup or dominance over all possible gSC configurations. The learned component's contribution is evaluated separately through the price and initialization ablations.

C.1 Independent direct-Gurobi regional reference

The retained development study also ran the original dense MILP on sixteen 256-cell regions using Gurobi 13.0.1, eight threads, a 1,800-second limit and relative MIP gap target 10^{-4}. Each run contains every finite directed route. At this granularity the source builder has one cell per node and all cells as route endpoints; there is no spatial aggregation, route restriction, learned field or block optimization. The retained logs show no user-supplied MIP start. These are direct solver runs on the physical formulation, not the nearest-neighbor planner ablation.

We checked the sixteen original result files and solver logs against their cached input road graphs and reference parameters. Table 8 reports their original incumbents and upper bounds, with the ordinary relative gap $100(U - L)/|U|$. The absolute denominator handles the negative-profit North Carolina case. All sixteen reached the time limit; the median reported gap is 0.890%. The historical solver tolerances and records are preserved; these incumbents have not received the separate physical-plan replay used for the national gSC plans. No new optimization or parameter selection was performed to add this reference.

These regional solver records characterize direct optimization at a scale where the dense model can be built. They do not establish the relative performance of recent learned-multiplier algorithms.

D Recent algorithms: common-LCO interfaces and complete results

The recent algorithms were published for physical problems different from LCO. Comparing their original objective values with ours would therefore conflate model and algorithm differences. We keep the LCO physical equations fixed and adapt each method's problem interface, as detailed below. Each uses one declared configuration; no parameter sweep or performance-driven rerun enters the reported comparison. Both implementations use the separately validated common physical compiler,

Table 8: Direct Gurobi on the full 256-cell development MILPs. These are existing solver outputs at the 1,800-second cap. Their paired comparison with the retained development price fields is in Table 25; the data are separate from the later frozen tests.

region	incumbent ($B/year)	upper bound ($B/year)	gap (%)
CA	2.0505	2.0553	0.232
IA	1.4505	1.4547	0.285
ID	0.4140	0.4195	1.308
KS	0.4766	0.4845	1.634
MI	1.5914	1.5966	0.324
MN	0.8875	0.8926	0.570
MO	0.2720	0.2801	2.886
NC	-0.0525	-0.0516	1.762
NE	3.3213	3.3334	0.361
NM	0.4765	0.4804	0.822
OK	0.0354	0.0395	10.367
PA	0.5967	0.6017	0.837
SD	1.5002	1.5144	0.942
TX	1.0236	1.0352	1.124
WA	0.7837	0.7871	0.434
WI	0.6712	0.6793	1.201

with kilotonnes and thousand dollars as solver units. Neither receives our national plan, learned field, spatial partition, import-cap formula, min-plus transformation or material-price floor.

D.1 Callback Benders from Chen et al.

The algorithm in Section 5 of Chen et al. (2026) separates integer decisions from continuous recourse and adds dual optimality cuts or Farkas feasibility cuts. Its first acceleration implements these cuts through Gurobi's lazy-constraint callback. We retain this algorithm: the master chooses LCO capital segments, and the recourse LP optimizes all remaining continuous variables on every allowed route. All-zero segments provide the initial feasible integer assignment; the recourse optimization produces its own starting plan.

The paper's second acceleration requires at least one collection center to open. LCO permits local application without processing, and its segment variables do not represent collection-center opening. That inequality would change the physical problem and is omitted. No substitute problem-specific inequality or heuristic from our method is introduced. This is a common-LCO algorithm adaptation, not execution of the authors' distributionally robust routing model.

Every case uses Gurobi 13.0.3, eight threads, a relative MIP-gap target of 10^{-4} and a 1,800-second optimization allowance including the initial recourse evaluation and callback work. Primal, dual and integrality tolerances are 10^{-9}. Recourse LPs use dual simplex; infeasibility information supplies Farkas cuts. Each job has 32 GiB allocated memory and a 24-GB solver cap. The sixteen runs execute on cluster CPU nodes; construction and the subsequent physical check are recorded separately. At a time limit inside a callback, the reported upper bound retains only bounds available before an unvalidated candidate could be accepted. Saved primal vectors come from solved recourse problems, not optimistic master objectives.

Small two-cell tests enumerate every binary assignment and verify that each optimality cut bounds every feasible recourse value and each feasibility cut excludes its generating infeasible assignment without excluding a feasible one. Direct and Benders optima agree under the solver tolerances, including zero-supply and missing-road cases. Every returned regional plan also passes the separate physical checker without rounding, repair or reoptimization.

The median Benders gap is 15.15%, versus 0.89% for the direct solver. The latter has better reported primal and dual values in all sixteen regions. The retained direct runs use Gurobi 13.0.1 and native units; their incumbents have not undergone the new independent physical replay. Solver-version, scaling and hardware differences prevent an attribution to the algorithm alone. Ordinary Benders

Table 9: All sixteen Benders runs and retained direct-Gurobi references. Plan and upper bound are in $B/year. Both methods reach their 1,800-second caps. Benders gap is $100(U-L)/|U|$; the North Carolina interval crosses zero, making its percentage large. Direct-Gurobi gaps are listed in Table 8.

	Benders (Chen et al.)			Direct Gurobi	
region	plan	upper	gap (%)	plan	upper
CA	2.0276	2.1150	4.13	2.0505	2.0553
IA	1.4342	1.5245	5.92	1.4505	1.4547
ID	0.4045	0.4788	15.51	0.4140	0.4195
KS	0.4567	0.5425	15.82	0.4766	0.4845
MI	1.5865	1.6880	6.01	1.5914	1.5966
MN	0.8836	1.0233	13.65	0.8875	0.8926
MO	0.2691	0.3607	25.40	0.2720	0.2801
NC	-0.0542	0.0052	1144.69	-0.0525	-0.0516
NE	3.3072	3.4165	3.20	3.3213	3.3334
NM	0.4759	0.9770	51.29	0.4765	0.4804
OK	0.0328	0.2644	87.61	0.0354	0.0395
PA	0.5921	0.6948	14.79	0.5967	0.6017
SD	1.4962	1.6442	9.00	1.5002	1.5144
TX	1.0094	1.1401	11.47	1.0236	1.0352
WA	0.7824	1.4299	45.28	0.7837	0.7871
WI	0.6650	0.7945	16.30	0.6712	0.6793

also retains the dense continuous transport recourse; these regional runs do not supply a national upper bound, and no intentionally oversized national allocation is counted as a performance failure.

D.2 Demelas et al.: graph predictor and independent LR interface

The framework of Demelas et al. (2024) takes an explicit MILP, its continuous-relaxation solution and a specified Lagrangian relaxation (LR). Its supplied network-design and assignment models do not encode LCO. We implement the published learner with a separate, classical variable-splitting interface to the complete LCO.

Equivalent bounded formulation. For each allowed original shipment x_{mij}, create sending and receiving copies s_{mij} and r_{mij}. The origin balance uses s, the destination balance uses r, freight is charged once to s, and $s_{mij} = r_{mij}$ restores the original shipment. To make the relaxed problems finite for arbitrary predicted prices, we derive explicit bounds from the original physical equations.

Let n be the number of cells, b_m the total exogenous supply of material m (zero except for raw manure), S_{mi} the storage quantity at cell i, and $K_o^{\max}$ the original per-cell capacity of operation o. Summing the original balances and using nonnegative consumption gives a bound on total storage of material m:

$$\sum_i S_{mi} \le \overline{S}_m := b_m + \sum_o n K_o^{\max} \max_s \alpha_{smo}, \tag{26}$$

where a maximum over absent conversions is zero. Thus $S_{mi} \le \overline{S}_m$ is valid for every original feasible plan. The upper limits on crop response and Eq.(26) give a finite upper bound $R_{\max}$ on total positive revenue. All other costs in the reference model are nonnegative. Let L_{init} be the objective of the explicit no-processing, no-transport plan using baseline crop yields and local storage and application. Every optimum is at least L_{init}, so, for the strictly positive freight costs in these data, an optimum satisfies

$$0 \le x_{mij} \le \frac{R_{\max} - L_{\text{init}} + \epsilon_{\$}}{c_{mij}}, \qquad \epsilon_{\$} = \$1. \tag{27}$$

The one-dollar allowance is numerical slack. These bounds preserve the original optimum: original feasible solutions below L_{init} need not be retained. They are objective-based bounds on the original shipment variables, distinct from our per-cell import cap and its mass-conservation argument.

Dualizing $s = r$ leaves the original local physical and integer systems coupled only through fixed objective coefficients. Our implementation passes the complete expanded model, with those equalities removed, to Gurobi; it does not call our spatial oracle or compress outside destinations. The

graph contains one node per variable and constraint, and the predictor outputs one multiplier per copied-flow equality. Every consistent original optimum lies in this bounded formulation, so any returned LR solver upper bound also bounds its profit. Small exact tests verify optimum equivalence, finite LR values and the subgradient support inequality. In the California test, retaining local integrality yields an LR bound below the continuous relaxation.

Architecture and features. We follow Appendix A of Demelas et al. (2024): variable features are objective coefficient, LP primal value, reduced cost and integrality; constraint features are right-hand side, LP dual value, equality flag and relaxation flag. Each node occupies its own four slots in an eight-dimensional vector, with the other slots zero. Less-than rows are negated into greater-than form. Graph edges follow the nonzero coefficient incidence, using unit weights and self-loops as in the public example runner. The graph convolution uses normalized adjacency. There are no geographic features or fitted input standardization from our model.

The input MLP maps eight features through width 250 to a width-500 representation. Five residual blocks contain nodewise layer normalization, linear graph convolution, dropout 0.25, and a 1000-wide MLP. The final representation splits into width-250 Gaussian mean and log-variance vectors, followed by a 250-wide decoder producing a correction to the LP multiplier. This has 6,450,751 parameters. We port the author's bounded log-variance sampler and standard-deviation-plus-epsilon normalization. A vector-norm implementation preserves the latter formula while giving a finite backward derivative for zero-valued node features. Sparse and dense versions of the graph operator agree in forward values and gradients on a numerical check.

This is a paper-level PyTorch implementation. The pinned example script uses 20 features and Adam, whereas the paper specifies eight features and RAdam; we follow the paper. Width 500 is the public model factory's default. We use RAdam at 10^{-4}, gradient clipping at five, and the published learning-rate decay settings. Thirty epochs, one seed and the final checkpoint are declared for this LCO comparison. All 360 training updates complete on the same twelve training regions; the four development regions are evaluated without checkpoint or hyperparameter selection. No test region supplies a training update. Activation checkpointing reduces memory use without changing the network.

Evaluation, costs and scope. Training takes 67.7 minutes on an A100 with 80 GB GPU memory, eight CPU threads and 192 GiB host allocation; peak allocated GPU memory is 51.24 GiB. The 256-cell expanded graphs contain approximately 1.58–1.64 million nodes. On the final sixteen test regions, median LP/graph preparation takes 36.0 seconds and median online evaluation takes 34.4 seconds. The latter includes graph loading, oracle construction, the LP-start reference, encoding and five stochastic learned predictions. These costs are not directly comparable to one spatial round. Every one of the eighty learned test-field oracles reaches optimal status, with a 120-second cap, eight threads and relative MIP-gap target 10^{-8} per evaluation.

The Demelas learner tightens its own LP-start bound in all sixteen cases, by a mean $\$0.017$M/year. Our learned bound before further multiplier optimization is lower in all sixteen, by $\$9.018$M/year on average. Our spatial relaxation at LP prices already improves on it by $\$6.038$M/year. Thus the external difference reflects the entire representation and relaxation, in addition to learning; it cannot be attributed solely to the neural architecture. The previously frozen within-method initialization study remains the controlled evidence for learned initialization. This new comparator uses the previously evaluated final-test cohort and is retrospective.

No direct national graph run is claimed. The expanded representation retains a variable for every candidate flow and adds its copy and consistency row; its size remains quadratic. Arithmetic memory estimates are distinguished from measured solver or allocation failures.

E Data, parameters and computational cost

Explicit national model storage. With $n = 63{,}873$ locations and eight materials, the all-pairs formulation has $N_x = 8n(n-1) = 32{,}637{,}570{,}048$ potential directed transport variables. Consider a compressed-column array representation of just these variables: each has two mass-balance coefficients, stored as two 8-byte values with two 4-byte row indices, one 8-byte column pointer, and three 8-byte entries for its objective coefficient and lower and upper bounds. These arrays require

Table 10: Complete recent learning comparison. All entries are upper bounds in $M/year, so lower is better even when negative. The Demelas column retains its best of five stochastic predictions; our columns use one round. Both LP-start references use their own method's LP representation and relaxation.

case	Demelas LP start	Demelas best of five	ours LP	ours learned
00	177.294	177.281	171.124	169.329
01	-4.710	-4.714	-4.865	-5.952
02	750.387	750.385	748.294	747.822
03	93.649	93.636	84.929	83.419
04	512.375	512.366	505.853	505.636
05	864.775	864.715	846.777	840.077
06	87.265	87.228	74.737	73.069
07	415.049	415.035	407.185	405.798
08	-0.839	-0.849	-1.077	-4.718
09	309.758	309.742	301.060	299.437
10	109.571	109.565	105.582	105.165
11	-7.096	-7.100	-7.116	-8.456
12	161.946	161.902	149.273	138.108
13	-20.106	-20.119	-20.817	-25.092
14	698.530	698.523	692.373	692.256
15	20.521	20.495	18.180	7.902

$56N_x + 8$ bytes, approximately 1.83 TB (1.66 TiB). Our largest cluster node reports 1,538,614 MiB (1.47 TiB) of memory; this is the nominal 1.5-TB node. The array estimate already exceeds this memory, before local decision variables, model-building objects, presolve, factorization or MIP search. This is a stated representation estimate, not a measured full-model allocation failure or a lower bound for every possible compressed or distributed implementation. It explains why we avoid materializing the dense MIP even on a large-memory cluster.

Historical diagnostic metric. The exploratory tables in this appendix and Appendix J also report gaps relative to the reference plan's value added: $100(U - L_{\mathrm{ref}})/(L_{\mathrm{ref}} - V_0)$. Here L_{ref} is the reference plan specified for each comparison, and V_0 is the best local no-processing/no-transport value. This diagnostic is distinct from the main text's national relative gap, $100(U-L)/U$, and from objective improvement over a planning baseline. Frozen regional comparisons use their prespecified bound differences in $/t.

Physical representation. For our internal physical-cell GNN and MLP controls, each input node represents one 10 km grid cell. The compact offset model uses the LP base and eight shared parameters; the external MILP-graph predictor has variable and constraint nodes (Appendix D.2). The input channels for the physical-cell controls are listed in Table 11; all inputs are available before the integer block solves used for learning. Supply and crop need describe material availability and demand; radius sums expose context beyond a single cell; LP duals summarize the continuous model's economic tradeoffs. These are design motivations, not claims that every channel is necessary. The features use outgoing road-distance rows, matching the regional implementation; the supplied matrix may be asymmetric.

Table 11: Node features before training-set standardization. The eight dual prices are signed; only nonnegative physical quantities use the logarithmic transform.

quantity	channels	transform
Raw manure supply (t/yr)	1	$\log(1+x)$
Total harvested area (km^2)	1	$\log(1+x)$
Crop N/P/K recommendation (t/yr)	3	$\log(1+x)$
Attainable crop value ($/yr)	1	$\log(1+x)$
Supply within 15/30/60 km (t/yr)	3	$\log(1+x)$
P recommendation within 15/30/60 km (t/yr)	3	$\log(1+x)$
Base LP material prices ($/t)	8	$x/10$

Each directed neighborhood has up to sixteen nearest road neighbors. Its two edge channels are $\log(1+d)/5$ and $1/(1+d/10)$, with d in kilometers. A two-layer SiLU MLP maps standardized inputs to width 64. At each of three graph layers, another MLP maps the concatenated neighbor state, receiving node state and edge channels to a message. The layer concatenates the node state with mean and maximum messages, passes them through an update MLP, adds a residual connection and applies layer normalization. A linear head returns eight material-price residuals. The cellwise MLP control uses the same input and output maps, with the three graph layers bypassed. It shares features, width and optimizer settings, but has fewer active layers and parameters; this comparison does not isolate message passing from network depth.

Training target and normalization. The retained source checkpoints were trained for 30 epochs with Adam at learning rate 2×10^{-3}, width 64 and three message-passing layers. An epoch visits every training region once in a seeded random order. Physical inputs use $\log(1+x)$, LP prices are divided by ten, and channel means and standard deviations are fitted on the training regions only. The output head starts at zero, giving an LP-price residual model. In each update, the source block solver supplies the net import/export subgradient g; the code backpropagates $g/\max(\|g\|_1, 1)$ and clips the parameter-gradient norm to five. This rescales each region's optimization feedback before updating the predictor. The export-minus-import feedback and its normalization are given in Eq.(6). Both source and corrected-objective models use this update mechanism, with their different training caps distinguished below. Autodifferentiation passes through the price transformation and predictor; the block solver provides the optimization feedback.

The original training block model also imposed an aggregate per-material import cap of total instance manure supply. On nonmetric roads, material can profitably leave and re-enter a block, so that aggregate cap need not preserve the dense optimum. The counterexample in Appendix I shows why the independent evaluation removes it and retains the valid per-cell caps of Eq.(2). Existing checkpoint outputs can still be evaluated with this block relaxation because the bound holds for arbitrary prices. Training and upper-bound evaluation are therefore distinguished explicitly. The completed corrected-objective cohort and its frozen confirmation are in Appendix F.

National feature approximation audit. A predictor trained on regional features can change behavior if national feature construction uses a different approximation. We isolate that implementation effect by holding the learned weights fixed. Regional feature construction sums manure supply and phosphorus demand over every cell within 15, 30 and 60 km. National prediction had used only the first 32 road neighbors for these catchments. A full-matrix replay of the original inference reproduces the saved price field to 4.8×10^{-7} absolute. Exact 15- and 30-km totals match, but 40,525 of 63,873 cells miss more than one tonne of 60-km supply, with a maximum omission of 5.59 million tonnes. Holding the checkpoint, LP field, normalization and post hoc material floor fixed, exact catchment features change 447,384 of 510,984 cell–material prices by more than 10^{-5}/t; the largest change is \$0.229/t. Any such price field yields a valid bound once the block model has per-cell import caps and considers every outside export destination; the feature shift limits claims about transfer quality. Table 13 compares the paired full reruns. Exact catchments tighten the learned upper bound by only \$83,185/year relative to the saved approximate-feature field, despite the widespread input change. This is a post hoc implementation sensitivity, not an independent confirmation of national learning.

Missing graph edges and the source-checkpoint audit. One national cell has no outgoing road neighbors. In the source implementation its sixteen padding entries, stored as -1, indexed the last node and introduced artificial messages. The supplied road matrix is directed: this cell still has incoming routes and appears in other cells' neighborhoods. The repaired aggregation masks absent edges in both mean and maximum operations and returns zero messages when no neighbor is available. Fully populated neighborhoods produce bitwise-identical outputs in a direct implementation check.

The final source-checkpoint audit combines exact catchments and masked neighbors, with the checkpoint, normalization, LP base, floor and partition fixed. This combined implementation was declared before its block bounds were evaluated. Table 12 retains the original field and both individual corrections. The mask repair alone changes 5,511 prices by more than 10^{-5}/t, but its \$361 aggregate-bound difference is small relative to solver accuracy. Both repairs together change the reported upper

bound by −$83,547/year. These small differences should be interpreted in light of solver tolerances; the substantive national conclusion is unchanged.

Table 12: Historical national implementation sensitivities in native units. All fields share the saved checkpoint and floor and use the full-destination per-cell-cap block relaxation. The combined correction fixes the implementation used in later studies.

implementation	upper ($B/yr)	change from saved ($/yr)	optimal blocks
Saved features and graph	135.334527	+0	2048/2048
Exact catchments only	135.334443	-83,185	2048/2048
Masked neighbors only	135.334526	-361	2048/2048
Both corrections	135.334443	-83,547	2048/2048

Applying the learned pipeline's additional three min-plus passes and the same floor to a zero neural residual returns the saved LP field bitwise unchanged. Its existing block bounds therefore provide an exactly processing-matched control. The corrected predictor tightens that bound by $705.8M/year in the original records. These implementation diagnostics retain the original solver tolerances; the current national comparisons use the stricter replay in Appendix G.

Input-processing replay cost. For the final input audit, float64 min-plus operations and exact catchment sums run on one H100 80-GB GPU. A directed nonmetric 97-cell test matches the CPU min-plus implementation exactly; sampled national catchments match to 2.9×10^{-11} absolute. The paired original/repaired national prediction, including exact catchments and distance loading, takes 41.2 seconds. This excludes the original sparse LP, training and the CPU block MIPs. The float64 GPU distance cache uses about 30.4 GiB; this replay accelerates the same quadratic calculation without changing its mathematical scope.

Table 13: Historical national feature sensitivity in native units, with the frozen network, LP field, material floor and 2,048-block partition. Every upper bound uses per-cell import caps and all outside export destinations; all block MIPs reached optimal status. Extra digits show the small paired change, which would be hidden by three-decimal billion-dollar rounding.

price field	upper bound ($B/yr)	gap ($B/yr)	gap / value added
LP field	136.040211	6.170628	74.289%
Learned, 32-neighbor catchments	135.334527	5.464943	65.793%
Learned, exact-radius catchments	135.334443	5.464860	65.792%

Parameters and data. The geographic inputs are GLW4-2020 cattle density and SPAM-2020 harvested-area and production rasters (FAO, 2024; International Food Policy Research Institute, 2024). The source notebook identifies the SPAM dataset DOI but omits the raster release; the derived inputs are included in the reproducibility package. Preprocessing converts density to animal counts, resamples counts, area and production by summation onto 10,000 m Equal Earth cells (EPSG:8857), and converts harvested hectares to km^2. Cell centroids select the contiguous-US cutout using 2024 Census state boundaries; Alaska, Hawaii and territories are excluded. Missing raster entries are represented as zero in the supplied inputs. The retained inventory models 29 crops.

The road notebook uses a local openrouteservice heavy-goods-vehicle matrix service, a January 31, 2026 OpenStreetMap extract and a 7,071 m snapping radius. The cached distances are converted from meters to kilometers and retained as directed values, with unavailable routes represented by missing entries. The cache preserves a mapping from retained cell geometries to original distance-matrix indices. Data years and reference economic parameters are distinct; these experiments do not estimate a contemporaneous national forecast.

Material properties and pyrolysis yields follow the reference inventory; manure supply uses a provisional 14.4 t per head per year; biochar earns a $200/t subsidy and a 10% yield bonus, both reference scenario parameters. The reference model includes the changes in Appendix A.9; the per-cell nutrient-coverage requirement is infeasible nationally because manure supplies 45% of nitrogen, 35% of phosphorus and 73% of potassium demand. The manure rate, plant capacities, subsidy and

yield bonus determine the reported dollar values. Road snapping and connectivity filtering omit 14,072 cells, holding 8.8% of input cattle and concentrated in the West. The national plan passes an independent implementation of the corrected formulation at 6×10^{-8} relative error; the block plans and regional bounds are checked by parity of the model builder with an audited implementation at 5×10^{-9} relative error.

Internal planner ablation. This internal comparison asks whether the selected facility segments or the final transport network explain a planning difference. It is separate from the published gSC baseline in Appendix C. The original national NN16 pipeline used 1,021,952 global transport arcs and yielded \$129.225B/year. Its final LP omitted some local block arcs, while the hub pipeline retained its hub arcs. We therefore reoptimize its frozen capital-cost segments on the union of global NN16, original local NN16 and every reference hub arc. This control uses 1,195,661 arcs and yields \$129.547B/year, versus \$129.870B/year for the hub plan on 1,129,511 arcs. Both NN16 variants and the hub plan pass the independent physical checker at maximum relative violation 5.96×10^{-8}. Exported accounts differ from optimizer objectives by less than \$830/year; all objective comparisons consistently use optimizer values. The stronger control takes 249 seconds to build, 1,371 seconds to solve, 1,039 seconds to export and 1,823 seconds to check on 32 CPU threads, using less than 40 GiB peak resident memory.

Table 14: Internal planner ablation. Gain is relative to the expanded NN16 control, with a common physical model and a superset of the hub plan's final arcs supplied to the control. Neither row is an external published baseline.

plan	profit (\$B/year)	gain (%)	transport arcs
NN16, shared union support	129.547	–	1,195,661
Hub segments (ours)	129.870	+0.249	1,129,511

Internal price-optimization controls. The regional comparison evaluates all price fields with the same per-cell-cap block model. *Subgradient*: one configuration starts from dense-LP duals and updates prices directly; the other starts from 32-neighbor LP duals and applies Eq.(4) before each bound evaluation, routing derivatives through its minimizing cells. Both use Polyak-type steps with their saved feasible-plan target. The five-update control is independently rerun with the corrected cap and reports the better of these two configurations. Each run uses an initial evaluation and five updates, for twelve bound evaluations across the pair. The source's longer runs had a 120-evaluation budget, but some stopped early and did not save their best price field; those incomplete artifacts do not establish a fully audited 16-region long-run comparison. Fourteen block evaluations in the short corrected reruns reach their time limit. Those runs retain finite upper bounds and use incumbent imbalances as heuristic update directions; no exact-subgradient convergence guarantee is asserted for them.

Bundle grid: the internal control uses a proximal cutting-plane scheme inspired by bundle methods (Lemaréchal, 1975; Frangioni, 2020). It evaluates $U(T(\pi))$, forms local affine models from its feedback, and solves a quadratic proximal subproblem in Gurobi. The initial weight matches a Polyak step toward a 2% reduction of the current objective; it is multiplied by five on a null step and halved on a serious step. Three initial weights (default, 0.1 and 0.01 times it) each receive 60 updates after the initial evaluation, and the best resulting field is reevaluated with the corrected block relaxation. Because $U \circ T$ need not be convex, this is a bundle-style heuristic, not an invocation of standard convex-bundle convergence guarantees. Its returned fields still give valid upper bounds.

Learned multipliers without the min-plus head: the model of Section 3.3 with the closure layer removed, so the predicted field is $\pi^{\mathrm{LP}} + \delta$ itself, trained and evaluated exactly as the full model under the same folds. This isolates the price transformation within our architecture. The MILP-graph encoder and problem-specific oracles of Demelas et al. (2024) are distinct; this ablation is not a reproduction of that published method. *Gurobi on the dense problem*: the original 256-cell instance with all 65,280 arcs, Gurobi 13.0.1 branch and bound with 8 threads to a 10^{-4} gap or a 1,800 s limit (the limit binds on every region); its incumbent is the best known plan of Table 25 wherever it beats the hub-network plan, and its final dual bound is the "Gurobi" column. In the corrected regional evaluation, block MIPs use a 10^{-6} relative gap and 120 s cap. A finite solver

upper bound is retained when the cap binds; that status is recorded rather than substituting the incumbent objective.

Cost. The original hub planner uses 64 block MIPs with 40-minute ceilings, run in parallel, followed by a national flow LP taking about 12 minutes on 64 threads. Computing the upper bound has a separate cost: price-input construction, prediction and 2,048 block solves. The stronger CG input takes 101 minutes; Appendix G reports the subsequent inference and solver budgets. Original training used 30 epochs on twelve regions (360 regional rounds), about two hours on one 8-core CPU node per fold. The corrected cohort uses four CPU threads. Neural training and block MIPs use CPUs; the final national price replay uses the H100 GPU described above.

Where the learned field tightens the national bound. For the retained sparse-input source checkpoint, Figure 3 plots, for each of the 2,048 national blocks, the change in the block bound from the LP-dual field to the learned field against the block's manure supply. By supply quartile (512 blocks each), the numbers of improved blocks and the net changes are 248 and $-$\$43.9M/yr in the lowest quartile (median supply 76 kt/yr), 313 and $-$\$144.3M in the second, 312 and $-$\$244.1M in the third, and 342 and $-$\$273.5M in the highest (median 1,272 kt/yr). The total reduction for this checkpoint is \$705.8M/yr; the largest individual worsening is \$2.21M. These values and the figure use the combined-correction field and full-destination block solves.

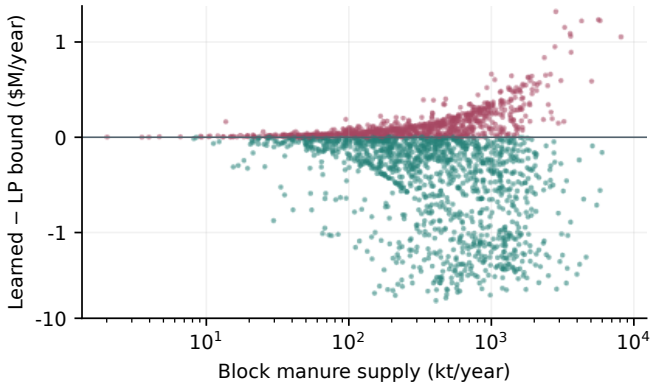


Figure 3: Source-checkpoint audit on the 32-neighbor LP input: change in the full-destination block bound from LP-dual to learned post-floor prices, against block manure supply (green: tightened; red: loosened). The horizontal axis is logarithmic; the vertical axis uses a symmetric-log scale with a linear interval between $-$\$1M and $+$\$1M.

F Frozen geographic confirmation and material-price controls

The regional studies separate prediction quality from the value of a learned initialization. The first reserved set compares predicted price fields. A subsequent follow-up tests direct optimization of the eight offsets on those same regions. A second reserved set then tests the initialization comparison with weights and optimizer settings fixed in advance. The three stages retain separate outcomes and protocols.

Training and checkpoint rule. We retrain with the per-cell-cap objective of Eq.(2), using the twelve fold-0 training regions and a 32-neighbor LP input. The other four original regions are development cases. A GNN and cellwise MLP each use seeds 0, 1 and 2; an eight-offset control uses seed 0. Every run performs 30 epochs, or 360 region updates. All training block solves reached their optimal status, so no updates were skipped. The declared checkpoint is epoch 30, irrespective of periodic development scores. Normalization uses only the twelve training regions.

The GNN has 93,576 active parameters; bypassing its graph layers gives a 6,024-parameter cellwise MLP. The offset model has eight parameters and no feature-dependent residual. Its base LP field

still encodes location-specific economics. The GNN/MLP use Adam at 0.002; offsets use Adam at 0.05, initialized at zero. All use the feedback scaling and gradient clipping described in Appendix E. Thirty training epochs take 54.7–61.8 minutes for the GNN seeds, 62.3–65.3 minutes for MLP seeds and 51.3 minutes for offsets on four CPU threads. These times exclude input construction, periodic development evaluation and final reporting.

Input-only geographic selection. We select sixteen new 256-cell closed-boundary regions. Eligible cells exclude all sixteen development footprints and New York, with a 100-km projected-centroid buffer. Positive-cattle anchors are shuffled with seed 20260922. Each region takes the nearest 256 eligible cells, has radius at most 180 km and requires at least 32 finite outgoing neighbors per cell. Selected regions exclude a further 50-km buffer from subsequent selections. The first sixteen admissible regions are retained without outcome filtering. Figure 4 shows the split. Their cells previously appeared in national planning and learned-price audits; the reserved objects are these new regional optimization contexts, not locations untouched by the entire study.

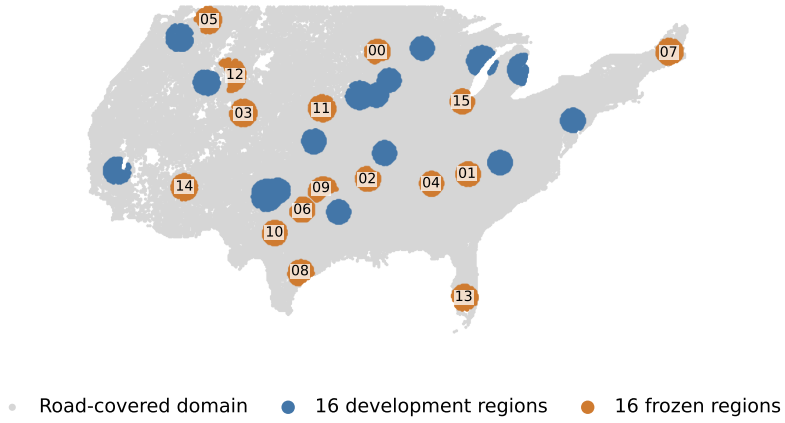


Figure 4: Development and reserved regional footprints. Selection uses geometry and physical inputs, with buffers specified in the text. The closed regional problems differ from their prior national context.

Frozen predictions and primary comparison. Data, code, weights, normalization and every price field for all sixteen regions are hashed before any candidate block-bound evaluation. The primary comparison is the seed-0 GNN against the processing-matched LP field, with no floor. Seeds 1 and 2, the MLP, the source sparse-input checkpoint and the offset model are additional declared controls. The primary per-region statistic is

$$\Delta_r = \frac{U_r(\pi^{\mathrm{LP}}) - U_r(\pi^{\mathrm{GNN}})}{\sum_{l \in r} b_l}, \tag{28}$$

in annual dollars per tonne of manure supply. Positive values mean a tighter bound, not increased achieved profit. We report the mean, median, all sixteen outcomes and a 20,000-resample paired regional bootstrap interval (seed 20260923). Regions are the sampling units; training seeds do not multiply the geographic sample size. Remaining spatial dependence and the input-selection distribution limit inference.

Block evaluation and classical comparator. To compare prices on the same mathematical objective, each field is passed to the block relaxation of Eq.(2). This bound-evaluation procedure is the optimization oracle used by training and refinement. Each distinct field receives independently constructed block models, four Gurobi threads, a 120-second block limit and relative MIP gap 10^{-6}. A cyclic evaluation order spreads load order across methods. The full-field classical control starts from the same LP base and uses five Polyak-type updates toward the no-processing value V_0: six rounds including the initial one. It differentiates through the minimizing cells of T and may reuse its own block models. A prospective amendment permits inexact incumbent feedback at a timeout

while always retaining solver upper bounds. No timeout occurred in this confirmation. This is a fixed short-run comparator, not a tuned optimum over all classical solvers.

A further amendment, made before prediction, adds the existing material floor to each seed-0 candidate and a separate floor-aware classical run. All unfloored primary outcomes are retained. Bitwise-identical fields reuse their recorded upper bounds with explicit links. The floor changes none of the candidate bounds in these sixteen regions; the classical runs also return identical best bounds. In total, 144 distinct predicted fields require 1,152 block solves, and the two classical configurations require 1,536 further block solves. All 2,688 reach optimal status.

Table 15: All reserved regional outcomes. Entries after V_0 are bound reductions relative to the same LP field, in \$/t (higher is better); neural and offset models use seed 0. The classical control uses six rounds. Every evaluated block is solver-optimal.

region	supply (Mt/yr)	V_0 (\$M/yr)	GNN (\$/t)	MLP (\$/t)	offsets (\$/t)	classical (\$/t)
00	7.538	1021.19	0.535	0.520	0.594	0.390
01	5.957	-9.64	2.109	2.386	2.147	2.087
02	9.200	-75.13	1.854	2.006	1.875	1.321
03	1.283	-11.67	1.692	1.784	1.723	1.674
04	5.756	399.77	0.637	0.789	0.687	0.719
05	1.527	355.18	0.107	-0.607	0.163	0.279
06	5.589	74.86	0.675	0.878	0.740	0.626
07	0.342	3.72	0.013	0.202	0.107	0.051
08	8.863	343.50	0.372	0.824	0.477	0.353
09	12.439	199.50	0.886	1.297	0.966	0.648
10	2.067	229.45	0.377	0.237	0.578	0.670
11	10.545	523.98	0.781	1.023	0.875	0.679
12	3.916	79.24	0.071	0.404	0.194	0.233
13	8.302	2609.44	-0.134	-1.097	0.197	0.000
14	0.540	-7.57	2.197	2.197	2.197	2.188
15	1.884	1049.84	1.958	1.911	2.444	1.663

Uncertainty and model complexity. The primary GNN's mean reduction is \$0.883/t, with interval $[0.509, 1.282]$, and its median is \$0.656/t; it improves 15/16 regions. Its other two seed means are \$1.015/t and \$0.939/t. The MLP seed means are \$0.922/t, \$0.430/t and \$0.841/t. The eight-offset model improves all sixteen regions, averaging \$0.998/t ($[0.632, 1.398]$). It beats the primary GNN in all sixteen paired cases; the mean difference is \$0.115/t ($[0.062, 0.183]$). These results support learned price calibration, without establishing a benefit from message passing.

The primary GNN and six-round classical control differ by only \$0.034/t on average ($[-0.058, 0.134]$). The offsets improve on that control by \$0.149/t ($[0.047, 0.271]$), with twelve wins and four losses; this is a secondary comparison. The declared two-evaluation portfolio $\min(U_{\mathrm{LP}}, U_{\mathrm{GNN}})$ beats the first two classical evaluations by \$0.784/t on average ($[0.458, 1.125]$). No retrospective best seed is used as the reported predictor.

F.1 Annual summaries and geographic coverage

Table 16 reports the first reserved cohort in annual and supply-normalized units. It remains separate from the final frozen initialization test. Figure 6 shows all three geographic cohorts; the outcomes were never pooled across cohorts.

Model size and computational cost. The eight-offset model improves all sixteen regions, averaging \$4.997M/year per region. Its eight parameters supplement the LP base. Without further multiplier updates, it also improves on the classical control's six rounds by \$1.076M/year per region on average. The corresponding supply-normalized difference is \$0.15/t ($[0.05, 0.27]$; secondary comparison). Median online time including shared LP setup is 12.1 seconds for offsets and 26.5 seconds for the classical control; offline offset training takes 51.3 minutes excluding validation and setup. Table 17 separates shared setup from online cost, which is dominated by block optimization.

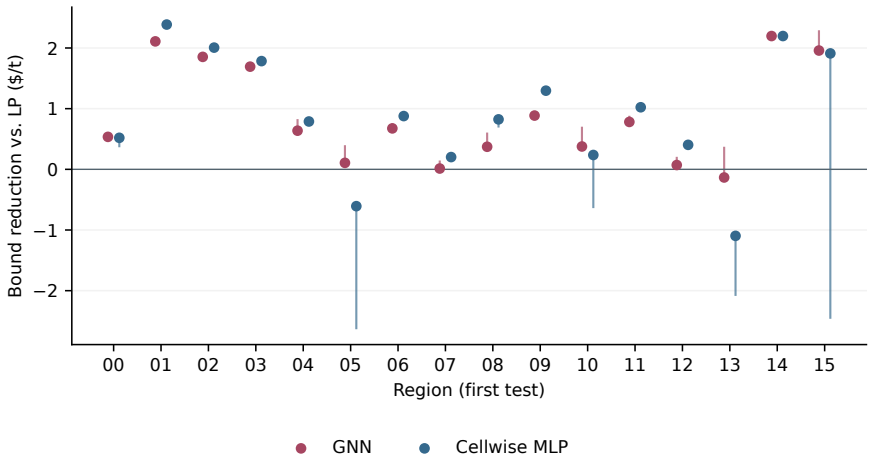


Figure 5: Neural price predictions on all sixteen regions in the first test. Dots denote seed 0; vertical lines span the three training seeds. Positive values indicate tighter LP-price bounds. No best-seed selection is used.

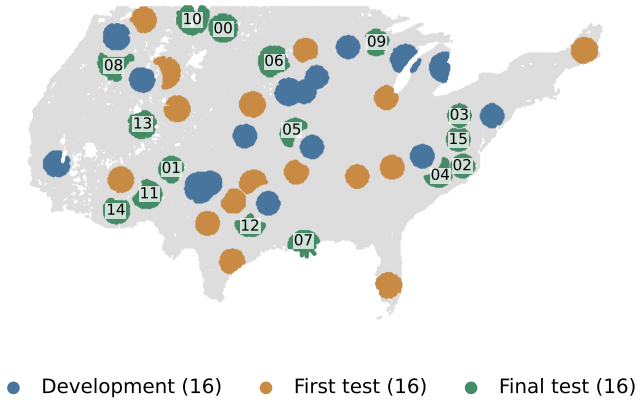


Figure 6: Geographic evaluation sets. The sixteen development regions provide twelve training and four development cases. The first and final test sets each contain sixteen new regional problems. The final set is at least 100 km from earlier footprints and New York. Numbers identify the final-test cases in Appendix F.3.

Table 16: Frozen regional price predictions: mean bound reduction relative to the same LP field, reported annually per region and normalized by each region's manure supply (higher is better). The 95% intervals apply to the normalized means and bootstrap the sixteen regions. Every learned field uses one round; the classical control uses six. The GNN seed-0 comparison is primary; other rows are declared controls. Detailed cases and seed variability are in the appendix.

field	mean ($M/year)	mean ($/t)	95% interval ($/t)	wins
GNN, seed 0 (primary)	4.425	0.88	[0.51, 1.28]	15/16
Cellwise MLP, seed 0	4.965	0.92	[0.44, 1.38]	14/16
Eight learned offsets	4.997	1.00	[0.63, 1.40]	16/16
Classical updates, six rounds	3.920	0.85	[0.53, 1.20]	15/16

Table 17: Median regional wall time. Online time includes prediction, fresh block construction and the stated evaluations. Shared setup includes the sparse input LP, V_0, features and initial construction. Training is excluded and reported separately. Medians of components need not sum to the median total.

method	bound rounds	after setup (s)	with shared setup (s)
LP field	1	4.55	11.80
GNN, seed 0	1	4.65	12.03
Cellwise MLP, seed 0	1	5.35	12.51
Eight learned offsets	1	4.70	12.11
Full-field classical updates	6	19.84	26.50

F.2 WHY A MATERIAL-OFFSET CONTROL IS TRACTABLE

The offset representation is useful both for prediction and for subsequent optimization: its dimension does not grow with the region, and its shifts commute with the price transformation. Let $\theta \in \mathbb{R}^{|\mathcal{M}|}$ add the same material-specific offset at every cell. Direct substitution into Eq.(4) gives

$$T(q+\theta)_{mB} = T(q)_{mB} + \theta_m. \tag{29}$$

This identity requires no triangle inequality. Once $P = T(q)$ is computed, the offset predictor needs only an array addition.

Proposition 4 and Eq.(8) give the convexity result and valid incumbent cuts in the main text; Appendix B.4 gives the proof and complete optimizer settings. The following specifies the matched optimizer and the original secondary comparison that preceded the final frozen confirmation.

Follow-up control. After inspecting the first confirmation, we declare a separate comparison within the same eight-dimensional offset class. A proximal bundle method starts at zero or at the frozen learned offsets and receives twelve rounds. Both runs use the settings in Appendix B.4, independently constructed blocks and within-run warm reoptimization with identical per-block solver settings. No parameter grid is selected from these outcomes. This secondary comparison measures initialization value and remains distinct from the frozen primary study.

Table 18: Secondary comparison in the same convex eight-offset class. Entries are mean bound reductions from LP prices in \$/t; higher is better. Each budget includes its initial evaluation. Both starts use the same bundle algorithm, without selecting a parameter grid.

initialization	1 round	2 rounds	6 rounds	12 rounds
Zero offset	0.00	0.25	0.86	1.03
Frozen learned offset	1.00	1.01	1.07	1.16

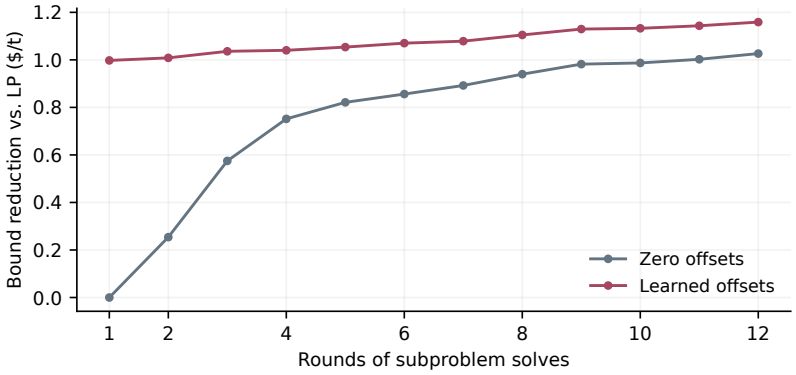


Figure 7: Mean bound reduction from LP prices in the secondary follow-up on the first sixteen regions. Both starts use the same offset optimizer, and the initial bound counts as the first evaluation. Equal evaluation budgets can require different wall times.

The learned start wins 15/16 paired cases at budgets two, six and twelve. At twelve, its mean advantage is \$0.132/t with regional bootstrap interval $[0.088, 0.184]$; at six it is \$0.214/t ($[0.114, 0.345]$). All block solves reach optimal status. Median online times for zero and learned starts are respectively 22.6 and 27.9 seconds at six evaluations, and 43.5 and 56.9 seconds at twelve, excluding the common input setup. Better oracle-budget performance therefore does not imply a universal wall-time speedup. Offline training cost is not recovered by the sixteen-region test alone; amortization depends on subsequent reuse and the required bound quality.

F.3 Additional frozen confirmation of material initialization

Before selecting these new cases, we fix the epoch-30 eight-offset checkpoint and every optimizer setting above. In material order $(\texttt{inf}, \texttt{infS}, \texttt{infL}, \texttt{infSPy}, \texttt{dig}, \texttt{digS}, \texttt{digL}, \texttt{digSPy})$, the offsets are approximately $(0.634, 1.871, 0.391, 8.396, 0.431, 1.190, 0.236, -1.295)$ \$/t. No further training or parameter selection is performed.

Input-only selection uses seed 20260923 and the same size, radius and connectivity rules, with a 100-km buffer from the original training/development footprints, New York and the first test cohort. New regions have 50-km mutual buffers. The first sixteen admissible regions are retained, supplying 0.559–11.897 million tonnes/year. These are new closed regional contexts; their cells appeared in the national study. Figure 6 shows both cohorts. Inputs, checkpoint, code and predictions are frozen before candidate block evaluation.

The primary endpoint is the learned start's mean per-tonne advantage over the zero start after twelve evaluations. The paired bootstrap uses 20,000 regional resamples and seed 20260924. Budgets one, two and six, runtime and solver status are secondary. All sixteen cases complete, and all $16 \times 2 \times 12 \times 8 = 3,072$ block solves reach optimal status. The primary advantage is \$0.094/t, interval $[0.056, 0.139]$, with fifteen wins and one loss; the median is \$0.082/t. The abstract reports the mean unnormalized difference, \$0.570M/year per region; this descriptive summary retains all sixteen cases and does not replace the prespecified per-tonne endpoint. All sixteen win/loss orderings remain separated when each run's best exact bound is bracketed by its recorded incumbent objectives and solver upper bounds; the signs are not explained by the recorded optimality gaps. No case or training seed is removed or substituted. At six evaluations the mean advantage is \$0.082/t ($[0.033, 0.140]$), with fourteen wins and two losses. The frozen offsets before further refinement improve every LP-price bound, averaging \$0.908/t ($[0.564, 1.278]$).

Table 19: Additional frozen confirmation: mean bound reduction from the LP field in \$/t (higher is better), using all sixteen regions. The primary comparison is the difference between starts at twelve evaluations. Initial evaluation counts are included.

initialization	1 round	2 rounds	6 rounds	12 rounds
Zero offsets	0.00	0.20	0.88	0.92
Frozen learned offsets	0.91	0.92	0.96	1.01

Table 20: Every final confirmation case. The final four columns are zero-start minus learned-start best upper bounds, divided by annual manure supply (\$/t); positive favors the learned start. Supply is in million tonnes/year and V_0 in million dollars/year.

region	supply (Mt)	V_0 (\$M)	1 round	2 rounds	6 rounds	12 rounds
00	3.317	154.07	0.541	0.617	0.403	0.167
01	0.559	-6.10	1.944	1.072	0.044	0.053
02	0.774	702.22	0.610	0.610	-0.118	-0.013
03	3.529	73.99	0.428	0.420	0.116	0.119
04	2.302	465.35	0.094	0.094	0.012	0.017
05	9.143	812.15	0.733	0.380	0.150	0.134
06	5.330	68.86	0.313	0.319	0.168	0.179
07	2.645	393.80	0.524	0.038	0.109	0.124
08	1.757	-7.65	2.072	1.837	0.025	0.013
09	3.076	282.70	0.528	0.528	0.102	0.035
10	1.926	92.03	0.216	0.180	0.026	0.100
11	0.609	-8.46	2.200	1.416	0.009	0.006
12	11.897	111.99	0.938	0.911	0.151	0.337
13	2.249	-25.57	1.901	1.816	0.042	0.045
14	3.831	571.00	0.031	0.031	-0.018	0.063
15	7.103	-3.79	1.447	1.245	0.097	0.127

At twelve rounds, median online times are 47.1 seconds from zero and 45.8 seconds from learned offsets, excluding shared input/setup (median 6.1 seconds) and offline training. Both starts use the same task and four solver threads. This descriptive result does not establish a universal speedup.

G NATIONAL PRICE INPUTS, ALL MODEL CONTROLS AND SOLVER BUDGETS

The national study tests whether corrections learned on regional instances remain useful at the full scale. To distinguish the effects of the LP input, learned correction and price floor, we evaluate their recorded combinations on the same block partition. The compact calibration, GNN and cellwise MLP use the same final seed-0 checkpoints as the first reserved regional study. Their per-cell-cap training uses twelve regions, thirty epochs and a 32-neighbor LP base. We retain every model's national output, irrespective of whether it improves the others. The GNN remains the primary predictor in the original cohort's comparison; the compact model is motivated by the first regional comparison and tested in sixteen further contexts with frozen weights and optimizer settings. National inference is an adaptive evaluation of one previously inspected instance, not an additional untouched geographic test.

Expanding the LP input by column generation. A sparse LP may miss profitable long-range routes and provide poor prices for the block relaxation. We therefore also test prices from an LP whose route set is expanded by classical column generation. The initial national LP relaxes capital-segment binaries and uses 32-neighbor transport arcs. Its variable count is 22.5 million. Classical arc generation prices every missing material–origin–destination flow by $\pi_{mB} - \pi_{mA} - c_{mAB}$ and adds the best positive-reduced-cost destination for each origin and material. We freeze the available completed round 5, after six LP solves, independently of the new candidate bound outcomes. It has 23,074,211 variables, objective \$135.110B/year and maximum remaining reduced cost \$56.55/t, so it has not converged to the dense LP relaxation. The recorded elapsed cost through this round is 6,081 seconds on 64 threads. Its dual prices are useful inputs; the reported MIP upper bounds come from integer block subproblems with all outside destinations, not that LP objective.

We apply the direct-cost transformation to a checked fixed point (maximum 64 passes) for the CG input and each residual output. Every observed fixed-point check has error below 10^{-9}. This stronger base is an input-transfer test for the three models trained on sparse LP prices. An earlier dense-input GNN checkpoint, trained with the aggregate-cap surrogate, is a separate control. Arbitrary checkpoint outputs can yield valid bounds through the per-cell-cap model, irrespective of their training objective. Its result does not replace the primary corrected-objective GNN or the compact model's independent confirmation.

Table 21: Every distinct national CG price field and the corrected 32-neighbor-input cohort. Raw/floored GNN and MLP outputs are identical for the corrected models; the floor also leaves both LP controls unchanged. Solver sums add retained block wall times, excluding aborted attempts and diagnostic runs; they are not elapsed national runtimes. The final source-checkpoint implementation sensitivities are retained separately in Table 12.

LP input	field	U (\$B/year)	optimal blocks	solver sum (s)
CG round 5	LP	133.109694	2047/2048	4629
CG round 5	Earlier dense-input GNN, raw	132.881788	2048/2048	3022
CG round 5	Earlier dense-input GNN, floored	132.218314	2048/2048	3022
CG round 5	Corrected-objective GNN	132.353136	2047/2048	3326
CG round 5	Corrected-objective MLP	132.413019	2046/2048	5623
CG round 5	Eight offsets, raw	133.741139	2046/2048	5183
CG round 5	Eight offsets, floored	132.154196	2046/2048	5182
32-NN	LP, matched processing	136.042481	2045/2048	3772
32-NN	Corrected-objective GNN	135.439260	2048/2048	1698
32-NN	Corrected-objective MLP	135.218237	2048/2048	2461
32-NN	Eight offsets, floored	135.459041	2048/2048	1753

Floor, feature and graph controls. All new national fields use exact radius catchments and mask missing graph neighbors. The inventory-derived floor changes one cell–material entry for the compact offsets and one for the earlier dense-input checkpoint. It changes none for the corrected GNN or MLP. Equal fields reuse recorded upper bounds only after exact array equality and hashes are recorded. The raw offset field is looser than the CG LP control; the specified floored deployment gives the tightest upper bound among these declared fields. The floor is a general material-value

rule applied to every cell, not an exception coded for the affected cell. Proposition 5 states its exact-objective guarantee and assumptions.

Numerical work. The paired CG inference uses 54.6 seconds on one H100 80-GB GPU task, including loading, CPU neural inference and float64 price processing; exact catchments are reused from the checked full-matrix computation. This excludes the input LP and block optimization. Each distinct field is evaluated cold in all 2,048 blocks with four Gurobi threads and a 300-second cap. Finite `ObjBound` values are retained at time limits. No missing block or incumbent objective substitutes for an upper bound. The field hashes, partition, per-block statuses and timing are retained with the aggregate tables.

Numerical verification of the fixed fields. The original national runs used Gurobi 13.0.1 in tonnes and dollars. A comparison of equivalent block formulations exposed sensitivity to the original tolerances: for one block, a tighter-tolerance feasible objective exceeded the previously reported upper bound by about $1,046. The discrepancy reproduced in versions 13.0.1 and 13.0.3. We therefore reevaluated all seven distinct CG fields and four sparse-input fields with one common representation: flows in kilotonnes, objective in thousand dollars, and each constraint row divided by its largest absolute coefficient when it exceeds one. Primal, dual and integrality tolerances are 10^{-9}, with `NumericFocus=1`, relative MIP gap 10^{-6} and an absolute gap target of $1. The replay uses Gurobi 13.0.3 and the same four threads and 300-second block cap. No physical coefficient, price, checkpoint or partition changes.

All national tables in the main text use this replay. Table 22 compares the retained original CG records with their reevaluation; all per-block records and residual diagnostics are supplied in the review package. The reported bounds remain subject to floating-point solver tolerances. The upper-bound inequality is exact under the stated assumptions; these solver outputs are not exact-arithmetic proof certificates.

Table 22: Numerical replay at unchanged national CG price fields. Differences include numerical sensitivity and changes in finite-time solver bounds. The rightmost column is replay minus original; lower upper bounds are tighter.

CG field	original U ($B)	replay U ($B)	change ($M)
LP prices	133.109462	133.109694	+0.231786
Earlier GNN, floored	132.218568	132.218314	-0.254324
Corrected GNN	132.353256	132.353136	-0.120734
Corrected MLP	132.413061	132.413019	-0.042028
Eight offsets, raw	133.740759	133.741139	+0.380653
Eight offsets, floored	132.153846	132.154196	+0.350057

We also reevaluate the already selected best field from each final regional zero-start and learned-start trace, without another optimizer step. All 256 resulting block solves reach optimal status. The mean advantage is $0.094244/t versus the original $0.094243/t; its paired regional-bootstrap interval is $[0.056001, 0.139297]$. Fifteen positive and one negative ordering remain separated using incumbent/bound intervals. Thus the reported frozen statistic is unchanged at its stated precision. The original study traces and runtime measurements remain intact; this subsequent numerical check is a separate cost.

G.1 Both LP inputs and environmental accounts

Environmental implications. The hub plan reduces modeled emissions by 17% and phosphorus excess by 59% relative to untreated local application (Table 24), although neither quantity is penalized in the reference objective. These changes are model-level consequences of the classical plan, conditional on the reference inventory and policy parameters.

Table 23: Internal price-learning ablation: upper bounds U in \$B/year, with matched transformations and the material floor. The CG reduction uses the LP-price reference; gap is $100(U - L)/U$. These rows vary inputs to the same spatial relaxation. All fields and solver statuses appear in Appendix G.

price field	32-NN U	CG U	reduction (%)	gap (%)
LP prices, no learning	136.042	133.110	0.00	2.43
GNN, seed 0	135.439	132.353	0.57	1.88
Cellwise MLP, seed 0	135.218	132.413	0.52	1.92
Eight learned offsets	135.459	132.154	0.72	1.73

Table 24: National environmental accounts, recomputed from exported plans. Excess is application beyond crop nutrient recommendations. The no-processing/transport reference and hub plan use the same inventory.

account	local untreated reference	hub plan
Emissions (Mt CO_2e/year)	462.5	383.3
Nitrogen excess (kt/year)	951	633
Phosphorus excess (kt/year)	237	97
Potassium excess (kt/year)	1,337	1,114

H Extended related work

Learning for mixed-integer programming. Many learned MIP components guide the search for feasible solutions. Our learning component instead targets the upper bound. Learned branching rules (Gasse et al., 2019), learned partial assignments that a solver completes, as in Neural Diving (Nair et al., 2020) and predict-and-search (Han et al., 2023), and learned search guidance (Khalil et al., 2022) speed up the discovery of good solutions. Learned branching can also change how quickly the solver tightens its dual bound; the solver maintains its validity. Our output is a price field, and by weak duality every price field yields a valid bound, so the learned component improves the upper bound on the attainable objective.

Learned Lagrangian multipliers. Demelas et al. (2024) predict multipliers from a MILP's variable–constraint graph and continuous-relaxation features, training on the Lagrangian bound with subproblem feedback. We build on that training principle. In the present application the full national variable–constraint graph is not instantiated. Our oracles retain physical cells, and material-offset coordinates preserve transport price differences while allowing convex refinement. The graph control uses the physical cell network. Exact outside-destination netbacks account for routes excluded by the sparse planner.

Learned bundle methods (Demelas et al., 2025), differentiable GPU optimization (Abbas & Swoboda, 2024) and classical warm starts from reduced problems (Cornielje et al., 2026) offer other ways to accelerate dual optimization. Le et al. (2026) analyze data-driven Lagrangian relaxation and learned warm starts; their guarantees do not by themselves establish rates for our specific oracle and refinement settings. Learning valid dual bounds in constraint programming (Bessa et al., 2025) shares the requirement that the solver's bound remain valid regardless of prediction quality. Decomposition-based decision-focused learning (Eilles-Chan Way et al., 2026) reduces training cost when predicting optimization inputs. Here the physical instance is given, and learning chooses multipliers used to bound its optimum.

Learning with a preserved guarantee. A learned prediction can be inaccurate, so an optimization method must specify which guarantees survive prediction error. In our construction, the block relaxation gives a valid bound for every multiplier field; the min-plus transformation affects tightness. Related methods preserve other types of guarantees: DC3 completes and corrects outputs to satisfy hard constraints (Donti et al., 2021). Self-supervised primal–dual learning jointly trains solution and multiplier networks using an augmented Lagrangian strategy (Park & Van Hentenryck, 2023). Our task is to choose prices for an explicit spatial relaxation, whose solver upper bounds cover a dense national model that is not instantiated.

Decomposition and aggregation bounds in operations research. Ma & Zavala (2022) combine uniform edge sampling with graph coarsening for supply-chain MILPs, including a manure-processing case. The sampled network gives a feasible original plan. Summing nodal material balances within groups, omitting internal freight and using minimum intergroup freight costs gives an upper-bounding relaxation. Cell-level technology decisions remain integer. Thus neither pairing a plan with a bound nor retaining facility integrality is a new principle of the present work. Our spatial relaxation instead retains local material balances and internal transport, and prices shipment consistency between blocks. Its empirical advantage over coarsening requires a comparison on the same physical model.

Pricing coupling resources to decompose a large problem is the price-directed idea of Adelman (2004), where dual prices of a relaxation guide the subproblems and bound the optimum; our price field plays that role for cross-block shipments. Lagrangian bounds for integer programs go back to Geoffrion (1974); Fisher (1981), and aggregation of variables or constraints to obtain computable bounds to Zipkin (1980a;b). Here we retain cell-level physical decisions and price the cross-block coupling, accounting explicitly for omitted routes. The instances we study have the integer structure of facility location with concave costs (Soland, 1974; Hajiaghayi et al., 2003): the capital-cost segments favor consolidation into larger plants, while freight and crop demand favor spatial dispersion.

Spatial optimization of manure and nutrients. The reference formulation in Appendix A treats facility siting, conversion, transport and crop application jointly. We study how to compute plans and upper bounds for this given physical model at national scale. Earlier manure and organic-waste logistics models coordinate transport, processing and reuse in spatial networks (Sharara et al., 2017; Sampat et al., 2019; Hu et al., 2019). Preuss & You (2024) study consequential and attributional poultry-manure LCO. Chen et al. (2026) develop a robust agricultural-waste collection and transport model with a Benders solution scheme. Their original physical formulations differ from the facility, conversion and crop-response equations used here; objective values across these formulations would not measure an algorithmic gain. Our focus is a feasible plan and a computable dense-model bound at 10 km resolution, including routes omitted by the planner.

I PROOFS OF FEASIBILITY AND UPPER-BOUND VALIDITY

The proofs follow the two outputs of the method. We first show that the sparse plan is feasible for the full model, then show that the block relaxation bounds its optimum. Further results justify the destination price representation, the LP starting point and the material-price floor.

Proof of Proposition 1. With z fixed, the physical model is linear. Let E be the final national LP's transport support. The concatenated block solutions satisfy every local physical constraint, exchange no material between planner blocks, and use only arcs in E. They are therefore feasible for the national LP with $z = z^\star$. Its optimum is at least their combined value. Extending its solution with zero flow on arcs outside E preserves every constraint of the dense program, so the checked value satisfies $L \leq \mathrm{OPT}$.

Proof of Proposition 2. Fix π and choose an optimal dense solution without positive-cost same-material transport cycles. Split each coupling flow as $x^\uparrow = x^\downarrow = x^\star$. (i) *Feasibility.* Restrict $x^\star$ to each B_k. Local constraints are preserved. The conversion graph is acyclic and every operation's output mass is at most its input mass, so at most $\sum_l b_l$ of a given material can be produced from the national manure supply. Because freight is positive, the same-material flow can be decomposed into simple paths after removing cycles. Each unit therefore enters any particular cell at most once, although it may enter a spatial block multiple times. The per-cell import caps in Eq.(2) are valid; an aggregate block cap would not follow from the supplied nonmetric distances. (ii) *Value.* In the expanded flow-copy formulation, each actual cross-block shipment contributes π_{mB} once positively and once negatively; freight is charged once. Summing block objectives therefore equals OPT. The aggregate-export formulation Eq.(2) replaces each actual destination coefficient by its best netback over *all* destinations outside the block. This cannot lower any block objective, so the resulting sum is at least OPT. (iii) *Boundedness.* $\mathcal{L}_k(\pi) < \infty$ because imports are capped and capacities are bounded. Maximizing each block can only increase its value, completing the inequality $U(\pi) = \sum_k \mathcal{L}_k(\pi) \geq \mathrm{OPT}$.

The following proves the main-text result that one multiplier per material and destination can replace independent arc multipliers without weakening the best attainable bound under the stated incoming-flow assumptions.

Proof of Proposition 3. For a fixed feasible block solution under π, assign each receiver's incoming copies to a cheapest admissible outside origin under μ. This leaves every physical constraint and per-cell cap unchanged, with the same import payment. On every sender copy, $\mu_{mAB} \geq \pi_{mB}$, so the export reward can only increase. Maximizing and summing over blocks proves the inequality. Conversely, assigning $\mu_{mAB} = \pi_{mB}$ embeds every destination field in the arc class, proving equality of the infima. Origin-specific import restrictions would invalidate this reassignment argument. The result concerns the full destination-price field, not the eight-dimensional offset subclass.

LP duals are a natural starting point because they already encode the continuous model's economic tradeoffs. The following result makes that connection precise when the full LP has been solved to optimality.

Proposition 6 (Block bounds at dense-LP dual prices)**.** *Let z_{LP} be the optimum of the full continuous relaxation and $\pi^\star$ its optimal mass-balance dual prices. For exact block optima,* $\mathrm{OPT} \leq U(\pi^\star) \leq z_{\mathrm{LP}}$.

Dual feasibility of every transport variable implies $\pi^\star_{mB} - \pi^\star_{mA} \leq c_{mAB}$. Relaxing all mass balances of the full LP at these prices separates the cell problems; transport has nonpositive reduced profit and may be set to zero. Strong LP duality makes the sum of these local relaxed optima equal z_{LP}. Now relax the internal mass balances of each LP-relaxed block using the same prices. Internal transport again has nonpositive reduced profit; each export has reduced profit $v_{mA}(\pi^\star) - \pi^\star_{mA} \leq 0$, and each import's price terms cancel. Dropping the import caps therefore bounds that block by the sum of its local LP optima. Summing proves the upper inequality; validity gives the lower one. This property motivates using LP duals, but requires duals of the full LP at optimality. The unconverged restricted national LPs in this study do not meet that condition. Their upper bounds instead come from explicit block solves, valid at any supplied prices.

Proof of Proposition 5. Take any optimal block solution at π'. If flooring increased an origin's export netback, its new maximizing destination must have price f_m; the new netback is therefore at most f_m, since freight is nonnegative. Replace that export by storage and non-crop application at its origin. This preserves the material balance and every plant and crop decision, and does not lower the block objective because $v_m^{\mathrm{loc}} \geq f_m$. Repeat for all affected exports. All remaining export coefficients are unchanged between the two fields. Returning imports to their original prices $\pi \leq \pi'$ can only improve the modified solution's value. It is feasible at π, so its optimum is at least the optimum at π'. Summing proves the result. □ Thus restricting a full price-field search to $\pi \geq f$ loses no exact bound quality. This is a statement about the full price field, not about convexity after clipping a restricted parameterization.

In US-ref-v1 the unrestricted outside-crop application variable a_{ml}^0 makes this option available. By the storage balance, $v_m^{\mathrm{loc}} = r_m^{\mathrm{mat}} - c_m^{\mathrm{sto}} - \beta_m c_m^{\mathrm{app}}$. The implemented floor uses the conservative value $f_m = r_m^{\mathrm{mat}} - c_m^{\mathrm{sto}} - c_m^{\mathrm{app}}$; every reference β_m lies in $[0, 1]$ and application costs are nonnegative, so $f_m \leq v_m^{\mathrm{loc}}$. This argument uses the stated objective without environmental penalties and does not require metric roads. It concerns exact block optima: time-limited solver upper bounds can still differ in tightness, so numerical comparisons retain their actual bounds and statuses. The analytic guarantee does not change the historical timing of the national floor's introduction.

Why an aggregate block cap is insufficient. A four-cell, one-material example shows the distinction. One tonne is available at a and earns \$200 if applied at d. The arcs $a \to b \to c \to d$ each cost \$1/t; every other directed arc costs \$100/t. Partition the cells as $\{a, c\}$ and $\{b, d\}$. The optimal dense flow earns \$197 and enters the second block twice, although no cell receives more than one tonne. For prices $(1, 2, 100, 101)$ at (a, b, c, d), the per-cell-cap block optima are \$1 and \$196, summing to the valid bound \$197. Adding an aggregate one-tonne import cap to each block changes those optima to \$1 and \$99. Their sum, \$100, is below the dense optimum. Thus the aggregate cap can invalidate the upper bound on a nonmetric network. This example was verified by independent small linear programs and does not require a neural model.

J ADDITIONAL RESULTS

The following studies examine classical optimization budgets, neural components and environmental accounts in more detail. Development comparisons remain distinct from the reserved regional tests; they help explain design choices but do not add independent test cases.

A stronger development-only bundle comparison. Additional retained development runs use a disaggregated proximal bundle with separate cuts for each block and six multipliers from 1 to 10^{-5} on an instance-dependent initial weight. The records contain 31–151 price-field evaluations per weight, counting the initial field; each run also solves no-trade block problems to seed its cut model. Selecting the most favorable weight in hindsight, the method first matches the earlier GNN's one-round bound after 6–24 price-field evaluations in nine of the sixteen development regions. The other seven remain unmatched in the recorded grid. These runs use the block relaxation with per-cell import caps and the original development-CV checkpoints. They do not add a frozen test sample or replace the matched eight-dimensional initialization comparison. Their costs also differ from the three-weight aggregated bundle grid in Table 25; the two grids must not be conflated.

Table 25: Development cross-validation bound quality by region (excess over the best known plan, % of value added; lower is better; value added in $M/yr). The best known plan is the better of the hub-network plan and Gurobi's incumbent on the dense problem after 1,800 s; "Gurobi" is the solver's own bound from that run. "Subgr. 5" takes the better of two five-update runs, twelve bound evaluations in total. The bundle grid uses three runs of 60 updates plus initial evaluations. Every reported price field is reevaluated using the block relaxation and per-cell import caps; finite solver upper bounds are retained when a block reaches its time limit.

Region	value added	LP prices	learned	subgr. 5	bundle	Gurobi
California	330.6	7.6	4.7	5.2	4.6	1.4
Texas	160.9	29.6	10.9	24.5	12.9	7.2
Nebraska	122.2	32.9	12.7	15.1	11.7	9.9
Washington	103.0	15.0	5.2	7.8	5.2	3.3
Iowa	83.0	48.3	7.2	11.1	6.3	5.0
Michigan	81.2	20.6	5.9	5.8	4.8	6.4
South Dakota	69.1	76.8	16.1	20.0	15.4	20.6
Idaho	67.1	49.1	24.1	27.6	26.3	8.2
Wisconsin	52.6	53.0	16.6	14.1	10.1	15.5
Kansas	45.6	88.1	28.9	30.2	26.3	17.4
Minnesota	41.9	51.1	13.8	14.6	10.2	12.2
Missouri	38.1	116.5	53.6	64.5	60.5	21.2
Pennsylvania	35.9	55.8	17.4	16.7	6.7	14.0
New Mexico	32.0	66.1	14.2	18.4	10.9	12.3
Oklahoma	13.8	194.0	49.0	56.9	55.2	22.5
North Carolina	3.6	403.1	68.3	88.7	72.0	22.7
median	–	52.1	15.2	17.6	11.3	12.2

Table 26: Audited component ablations on all 16 development-CV regions. Every field uses the corrected block relaxation with per-cell import caps. Gaps are percentages of the same best known plan's value added. The no-LP-input model removes eight feature channels but retains LP prices as the output base; the no-message-passing model uses a per-cell MLP; the no-min-plus model outputs raw residual prices. All block evaluations are solver-optimal except five in the no-min-plus variant, for which valid solver upper bounds are retained.

variant	improved over LP	median gap	worst gap
No LP-dual input	16/16	13.8	71.3
No message passing	16/16	14.1	52.0
No min-plus head	16/16	15.4	64.5

Representation and valid component results. The no-message-passing control often matches the larger graph model in the audited development cases. It differs in active depth and parameter count,

so this alone does not isolate the causal effect of messages. On the national CG input, material-wide means explain 99.926% of the corrected GNN residual's squared norm, versus 26.885% for the MLP residual. This describes corrections, not the spatially varying LP base. It motivates the compact class but does not prove an oversmoothing mechanism. The reserved comparisons report all models and separately confirm the eight-offset initialization.

Environmental-account breakdown. Table 24 reports the aggregate national emissions and nutrient excesses. Application and storage emissions fall by 107.8 and 58.7 Mt CO_2e/year, while processing, freight and capital add 85.4, 1.1 and 0.9 Mt. Applied N falls 12.6%; applied P and K each change by less than 0.3%. These are reference-model accounts, not estimated realized impacts. A quarter of applied nitrogen remains in excess; the objective places no penalty on it.