CERTIFICATION, NOT SEARCH: DEPLOYMENT-TIME MODEL SHARING UNDER HARD PER-CELL RISK CONTRACTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deployment under hard per-cell risk contracts must decide which trained predictors to store, which cells may share them, and when local fallback is required, while minimizing serialized model, payload, and routing cost. Existing procedures commit to a sharing structure during training: multi-task models fix parameter sharing, mixture-of-experts learns routing, and clustering fixes partitions before deployment. Deployment still requires a separate decision: whether a sharing structure can be certified against a per-cell risk budget from the calibration data actually available. We propose Certified Shared Decoder Selection (CSDS), a deployment-time framework that builds a finite candidate portfolio, certifies every cell–action pair from calibration data, and selects the shared subset and the per-cell assignment minimizing a declared additive coding-cost proxy over the certified action set. Because the contract is cell-separable, certifying the individual cell–action pairs certifies every deployment plan simultaneously. This property of the contract, rather than of the testing procedure, is what makes a deployment portfolio certifiable at all: testing the plan family directly yields a bound looser than the entire risk budget, certifying nothing. Experiments on five 30-seed tabular contract settings, predictor-family controls, and a four-domain transformer-adapter deployment show that CSDS reduces deployment cost against both all-local deployment and a calibration-tuned clustered router, with no observed contract violation. Matched controls holding the candidate set and fitted predictors fixed identify certification, not subset search, as the scarce resource, while the certificate family and its failure-budget allocation remain live deployment decisions. Code is provided in the supplementary material.

1 INTRODUCTION

Modern deployment serves heterogeneous subpopulations under hard contracts while operating under fixed memory and communication budgets. A medical system may require separate guarantees across demographic groups, and an edge fleet may need different predictors for different environments while sharing parameters whenever possible. Such contracts partition data into contract cells, each requiring a predictor for a downstream task. A local predictor pays serialized cost in every cell, while one global predictor pays once but may violate a cell-specific risk budget. The binding question is which sharing decisions can be certified from the calibration data a deployment actually has.

Existing approaches treat sharing as a modeling choice. Multi-task learning assumes a predefined sharing structure (Ruder, 2017; Zhang & Yang, 2022), mixture-of-experts learns routing during training (Shazeer et al., 2017; Fedus et al., 2022), and clustering fixes partitions before deployment (Jacob et al., 2008; Ghosh et al., 2020). These methods generate useful predictors but do not answer the deployment-time question: given candidates and hard per-cell contracts, which predictors should be stored and which cells should reuse them? The selected predictors determine coverage, the cell assignment determines payload cost, and the certificate determines which actions are admissible. Average accuracy is insufficient: a model can improve the mean and still violate a protected cell, while a local solution can satisfy every contract at prohibitive serialized cost.

Certified Shared Decoder Selection (CSDS) separates candidate generation, finite-sample certification, and cost-aware portfolio selection. Each predictor acts as a decoder for its cells' label stream and

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.

is charged the code length it induces. Candidate generation produces global, shared, and local actions. Certification attaches a valid risk bound to every cell–action pair, local fallback included, so that no action escapes the contract. Selection minimizes the coding-cost proxy over the certified action set. Sharing therefore becomes a deployment variable rather than a property fixed by the training architecture. The structure that makes this tractable is cell-separability: the space of deployment plans grows exponentially with the number of contract cells, yet certifying the individual cell–action pairs certifies every plan at once, a property of the contract rather than of the test used to certify it. Testing the plan family directly instead yields a bound looser than the entire risk budget, certifying no action at all and leaving the deployment problem with no feasible solution.

On five 30-seed tabular contracts, predictor-family controls, and a four-domain transformer-adapter deployment, CSDS lowers cost against all-local deployment in every paired run and against a calibration-tuned clustered router in almost all of them, with no observed contract violation. Holding the candidate set and the fitted predictors fixed identifies certified candidate coverage as the source of the deployment saving: the margin against the router grows monotonically with the certified pool, while exchanging the exact solver for a calibration-selected fixed-K policy over that same pool leaves the margin unchanged. Certification, not search, is the scarce resource, and it remains a live deployment decision: its price shrinks as calibration support grows, a tighter bound recovers part of that price at the same failure level, and reallocating the same failure budget across candidates from the tune split enlarges the certified action set and lowers deployment cost. Our contributions are:

- **Problem formulation.** We introduce certified shared decoder selection, a deployment-time optimization problem that determines a candidate-constrained sharing portfolio under hard per-cell risk constraints and an explicit serialized-cost budget.
- **Certification structure.** We show that cell-separability collapses the certification multiplicity from the exponentially many deployment plans to the individual cell–action pairs, which is what makes the portfolio certifiable at deployment scale, and we characterize the resulting cost path, the certificate price, and the sharing boundary.
- **Algorithm.** We develop a cost-aware portfolio method combining candidate generation, finite-sample certification, and cost-aware selection, and show that allocating the failure budget from the tune split enlarges the certified action set at the same failure level.
- **Controlled attribution.** We isolate where the saving comes from: certified candidate coverage, not subset search, with a dose–response over the candidate budget and a matched-candidate control that separates the certified pool from the selection layer.

2 Related Work

Parameter sharing and modular multi-task learning. Multi-task learning studies hard and soft parameter sharing (Ruder, 2017; Zhang & Yang, 2022), while modular methods compose adapters and experts (Rebuffi et al., 2017; Houlsby et al., 2019) and mixtures route among them (Yuksel et al., 2012; Shazeer et al., 2017; Fedus et al., 2022), with recent work fusing task-specific modules (Tang et al., 2024; Wang et al., 2024). All of these fix sharing structure during training. CSDS instead makes certified sharing coverage a deployment variable and treats routing as a discrete, certified cell assignment. Our baselines instantiate this family at deployment: an all-local portfolio, a clustered router fitting one predictor per KMeans cluster, its count chosen on calibration, a certified mixture-of-experts router with a learned gate (Shazeer et al., 2017), and the best fixed-size portfolio on identical candidates. All methods share the fitted predictors: CART on the tabular contracts and LoRA adapters (Hu et al., 2022) on a frozen DistilBERT (Sanh et al., 2019) for the neural deployment.

Description length and deployment cost. Minimum description length treats the model as part of the code for the data (Graves, 2011; Grünwald, 2007; Blier & Ollivier, 2018), and compression and quantization make serialized size an explicit design variable (Han et al., 2016; Jacob et al., 2018). We adopt that accounting and couple the serialized model cost to a certified cell-assignment decision taken at deployment time rather than fixed during model training.

Risk control, selective prediction, and subgroup guarantees. Conformal prediction and its risk-control descendants give distribution-free finite-sample guarantees for post-processed predictors

(Vovk et al., 2022; Bates et al., 2021; Angelopoulos et al., 2025; 2024), and later work relaxes exchangeability under shift (Tibshirani et al., 2019; Barber et al., 2023; Farinhas et al., 2024). CSDS applies that discipline to contract cells and shared model portfolios. Abstaining rather than predicting is the standard way to buy accuracy on the served stream (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017). Our referral action plays that role per contract cell. Requiring a guarantee on every identifiable subgroup rather than on average is the concern of multicalibration and subgroup fairness (Hébert-Johnson et al., 2018; Kearns et al., 2018). CSDS takes the subgroups as given and asks which shared predictors can be certified on them under a serialized-cost budget.

Risk-controlled selection and deployment-time sharing. Risk-controlling selection and Pareto testing optimize secondary costs under explicit risk limits (Laufer-Goldshtein et al., 2023a;b). Deployment-time sharing has systems precedent (Jiang et al., 2018). CSDS couples per-cell certificates with subset selection, cell assignment, and serialized cost.

Tabular evaluation. Recent tabular benchmarks show that random rather than time-based splits can change model rankings (Rubachev et al., 2025); we use contract-first splits throughout.

3 Methodology

3.1 Deployment Contracts

Let $C = \{1, \dots, m\}$ be contract cells and $H = \{1, \dots, K\}$ shared predictor candidates. For cell c and predictor h, let $L_c(h)$ be calibration payload, $L_c^{\text{test}}(h)$ test payload, $\text{UCB}_c(h)$ a finite-sample risk upper bound, and $b(h)$ serialized model cost. Let $\text{local}(c)$ denote local fallback. For the classification contracts used in this paper, $L_c(h)$ is the negative log-likelihood code length in bits on the calibration samples of cell c under action h,

$$L_c(h) = -\sum_{i \in \mathcal{I}_c^{\text{cal}}} \log_2 p_h(y_i \mid x_i),$$

and $L_c^{\text{test}}(h)$ uses the same coder on held-out samples. Thus bits per test sample is the serialized-cost proxy divided by the number of test samples. For CART, $b(h)$ is computed from the serialized node and leaf counts of the fitted tree rather than the maximum capacity of its depth. We use expected 0-1 error for $R_c(h)$. Payload remains NLL code length. The protected field that carries the cell resolution is withheld from every predictor's input, for CSDS and for every baseline, so no predictor can recover a sample's cell or specialize to it implicitly (Appendix A.3.1). We optimize a declared additive coding-cost proxy: it prices payload, model and routing bits, and excludes architecture metadata, quantization specifications, encoder overhead and cell-to-index mappings, which are common to every compared method and therefore cancel in every comparison we report.

Each predictor acts as a decoder for the label stream: the payload of a sample is its code length under that predictor. The contract is selective: every cell must receive either a certified predictor or an explicit referral. A cell is served by the deployment only if its local predictor is certified, $\text{UCB}_c(\text{local}(c)) \le \tau$; otherwise it is referred, at a declared per-sample cost κ and with no prediction issued. Referral is a contract action, not an exemption: the guarantee therefore covers every contract cell, and the served fraction is a reported deployment quantity rather than a filter applied before measurement. Requiring a certified local action on every served cell restricts the policy class, and buys a property the optimization relies on: every subset $S \subseteq H$, including $S = \varnothing$, then admits a feasible assignment on every served cell, so Theorem 1 is well posed over the whole subset lattice. The price is that an uncertified local predictor forces referral even when a shared candidate would be certified. This binds on 1.1 to 1.8 percent of contract cells on Covertype and 7.6 percent on Electricity (Appendix A.3.1), so it is the clearest direction in which the formulation could be loosened. Local fallback is a certified action for every served cell, and these m admission tests are carried inside the same simultaneous certificate as the shared actions rather than on a separate budget.

The decision variables are a shared subset $S \subseteq H$ and an assignment $a(c) \in S \cup \{\text{local}(c), \text{refer}\}$. The deployed assignment code is a fixed-width routing index: every cell uses $\lceil \log_2(|S| + 1) \rceil$ bits to identify one shared predictor or the local-fallback tag. Thus

$$A(S, a) = \sum_{c \in C} A_c(S, a(c)), \qquad A_c(S, a(c)) = \lceil \log_2(|S| + 1) \rceil. \tag{1}$$

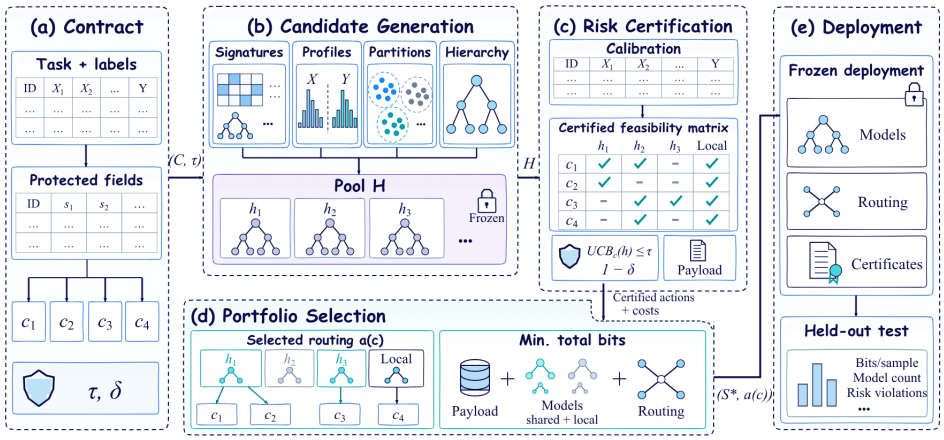

Figure 1: **CSDS architecture.** (a) Protected fields define contract cells and a risk budget (τ, δ). (b) Candidate generation forms a pool H, frozen before any calibration label is read. (c) Calibration certifies every cell–action pair, local fallback included, giving a feasibility matrix and payload costs. (d) Selection minimizes payload, model and routing bits over the certified set. (e) The frozen portfolio is scored on the held-out test data that no earlier stage has seen.

This cell-separable form is the routing code used in all experiments. The objective is

$$\begin{aligned}\min_{S,a} \quad & \sum_c L_c(a(c)) + \sum_{h \in S} b(h) + \sum_c b_{\text{local}}(c)\, \mathbf{1}[a(c) = \text{local}(c)] \\ & + A(S, a) + \kappa \sum_c n_c\, \mathbf{1}[a(c) = \text{refer}],\end{aligned} \tag{2}$$

subject to

$$\text{UCB}_c(a(c)) \le \tau \quad \text{for every cell with } a(c) \neq \text{refer}.$$

The reported metric is test bits per sample, including payload, model, and assignment terms. The proxy includes all modeled deployment terms. Payload bits describe the per-cell residual stream, model bits describe the serialized shared predictors, and assignment bits describe the routing table. Local fallback adds its own model and assignment terms. A shared predictor is useful only when its payload savings exceed these deployment costs while every assigned cell remains certified.

The same objective can be instantiated with other additive system costs such as latency, memory, energy, or FLOPs; the theory uses only additivity over cells and actions. The risk budget τ is a contract-level quantity set by a regulatory requirement, a service-level objective, or a predeclared standard. The certificate uses calibration data only, and the final test payload is reported only after the shared subset and the per-cell assignment are both fixed.

3.2 Method Overview

CSDS has three stages: candidate generation forms a finite portfolio from clustering, feature and target profiles, and baseline partitions; certification assigns simultaneous finite-sample upper bounds; selection minimizes the calibrated deployment objective. Each predictor adds serialized bits but may reduce payload on several cells, and local fallback preserves feasibility on every served cell when no shared action is certified there (see Figure 1 and Algorithm 1).

3.3 Candidate Portfolios

The implementation forms five certified candidate views that play complementary roles. KMeans clusters of local model signatures, seeded as in Arthur & Vassilvitskii (2007), group cells whose local predictors are similar. KMeans on feature-space profiles and on target profiles identifies cells that may share response structure even when their local models differ. A baseline-compatible KMeans partition

Algorithm 1 Certified Shared Decoder Selection

Require: contract cells $\mathcal{C}$, disjoint fit/tune/calibration splits, risk budget τ, and failure level δ
Ensure: selected subset $S^\star$, assignment $a^\star$, and certificates

1: **Fit split:** build the candidate pool from local signatures, feature and target profiles, baseline partitions, and the hierarchy, and fit every candidate.
2: **Tune split:** select the deployed candidate family $\mathcal{H}$, which fixes $K = |\mathcal{H}|$ and the multiplicity $M = m(K + 1)$. *No calibration label has been read up to this point.*
3: **Calibration split:** compute $L_c(h)$ and $\mathrm{UCB}_c(h)$ at level δ/M for every $c \in \mathcal{C}$ and every $h \in \mathcal{H} \cup \{\mathrm{local}(c)\}$.
4: Admit cell c iff $\mathrm{UCB}_c(\mathrm{local}(c)) \le \tau$; refer the remaining cells.
5: **for** each feasible shared subset $S \subseteq \mathcal{H}$ **do**
6: Assign each cell to $\arg\min_{a \in S \cup \{\mathrm{local}(c)\},\, \mathrm{UCB}_c(a) \le \tau} \mathrm{cost}(c, a)$.
7: Evaluate the deployment objective $J(S, a)$.
8: **end for**
9: **return** $S^\star = \arg\min_S J(S, a)$ and assignment $a^\star$.

preserves the comparison family, and a deterministic average-linkage hierarchy provides a sequence of coarser sharing levels. Each candidate is refit on its training cells. The deployed portfolio retains flat predictors and can add non-flat predictors selected on a tune split, which fixes the candidate set before calibration. Extra candidates are ranked by marginal payload savings net of serialized-model and routing cost under the same final deployment objective that selection later minimizes.

Proposition 3 gives a sufficient condition for dominating a baseline family; the deployed portfolio is capped at twelve columns, and empirically the CSDS calibration objective is no larger than the router objective in 145 of 150 headline runs (Appendix A.3.15). The remaining views add complementary actions, and tune-time ranking decides whether they are retained. Appendix Table 22 reports a candidate-budget control in which the gain grows with the budget, and the candidate-reference gap stays within 0.19 bits/sample of an exhaustive partition-induced reference on small instances, and ahead of it on three of the five contracts, so the deployable portfolio loses little to exhaustive search.

3.4 Certified Portfolio Selection

We use a finite-sample Hoeffding risk bound (Boucheron et al., 2013) over all candidate-cell pairs:

$$\mathrm{UCB}_{c,h} = \hat{R}_{c,h} + \sqrt{\frac{\log(2M/\delta)}{2n_{c,h}}},$$

where M is the number of tested cell-action pairs and $n_{c,h}$ is the calibration sample count. A cell chooses from $H \cup \{\mathrm{local}(c)\}$, so the tested family is the mK cell–candidate pairs together with the m local admission tests, and the multiplicity is $M = m(K + 1)$ (Proposition 1). Charging the local tests to the same budget is what makes local fallback a certified action rather than an uncontrolled escape from the contract. Both K and m are fixed on the tune split before any calibration label is read, giving M between 1536.0 ± 125.8 and 1604.3 ± 93.5 on the Covertype contracts and 545.7 ± 140.1 on Electricity; the admission step never reduces it. All feasibility and portfolio decisions use calibration data, while the candidate family itself is fixed on the tune split. Selection minimizes a calibrated deployment objective, while certificates constrain contract risk rather than payload or routing generalization. Test data are used only for reporting. For a fixed shared set, each cell chooses the cheapest feasible action, taking local fallback when no certified shared action is cheaper than storing a local model. Headline results use the two-sided Hoeffding bound required by Proposition 2; one-sided bounds certify more cells and are evaluated in the certificate-family comparison. Section 4.3 prices the certificate family. The exact solver enumerates subsets for the headline budget, and greedy and coverage solvers provide larger-budget variants.

3.5 Sharing Boundaries

A deployment plan is a pair (S, a), so the plan space has size $2^K(K + 1)^m$. Risk-controlled selection controls error over the family it tests, and the deployed configuration must belong to that family (Angelopoulos et al., 2025; Laufer-Goldshtein et al., 2023b;a), so testing plans directly pays a

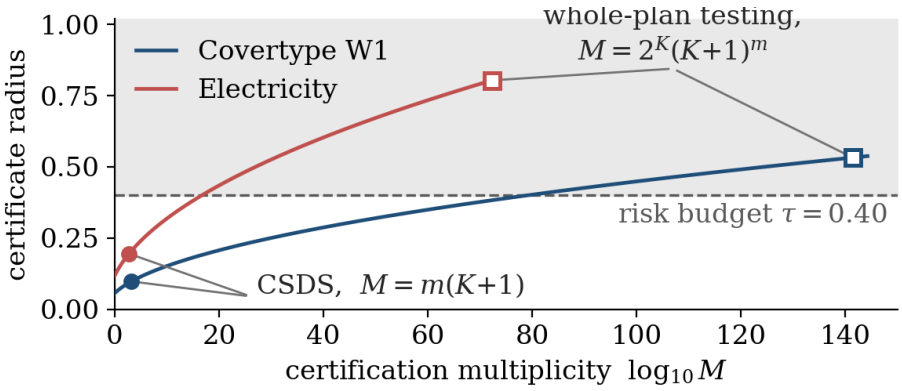

Figure 2: **Cell-separability is what makes the portfolio certifiable.** Two-sided Hoeffding radius against certification multiplicity at $\delta = 0.05$. The $m(K+1)$ cell–action primitives (circles) leave a wide margin below the contract budget τ; testing the $2^K(K+1)^m$ plans directly (squares) pushes the radius past τ, the shaded band in which no action certifies at any empirical risk.

multiplicity exponential in m. Testing only the primitive cell–action pairs is cheap, but no plan is then itself a tested hypothesis; validity for the deployed plan comes from the contract, not the test.

Proposition 1 (Certification multiplicity)**.** *The constraint and the objective are both cell-separable, so it suffices to certify the $m(K+1)$ cell–action pairs once. On the certificate event every plan built from certified pairs satisfies the contract simultaneously, so the multiplicity is $m(K+1)$ rather than the $2^K(K+1)^m$ deployment plans that those pairs jointly certify.*

The gap is not a constant factor. Testing the plan family directly gives radii of 0.53–0.55 on Covertype and 0.80 on Electricity, against 0.10 and 0.19 at multiplicity $m(K+1)$; these exceed the risk budget $\tau = 0.40$, so whole-plan testing certifies nothing at any empirical risk (Figure 2).

Proposition 1 is a property of the contract rather than of the test: cell-separability transfers validity from the certified primitives to every plan composed from them, so the simultaneous bound can be any valid one while the multiplicity stays $m(K+1)$. This is what makes a deployment portfolio certifiable at scale, and it also determines how the portfolio is searched.

Theorem 1 (Exact portfolio selection)**.** *For fixed S, the objective decomposes over cells. Independent per-cell minimization is globally optimal and costs $O(m|S|)$ evaluations. For finite H, exact enumeration over feasible subsets minimizes the deployment objective over all $S \subseteq H$ and all feasible assignments of contract cells to their certified actions.*

Theorem 1 splits the problem into a hard part and an easy one: choosing S is combinatorial, but the assignment given S is free. On the evaluated contracts, subset search does not bind. A policy that simply keeps the $K^\star$ cheapest certified columns attains the exact optimum on every contract we study (Section 4.4), because the cost path is flat near its minimum. The formulation characterizes the admissible portfolios; the flat cost path means practitioners need not run a combinatorial solver on contracts of the size we study here, where the candidate pool stays at or below sixteen columns.

Next, how model cost changes the optimum. Let $P(S,a)$, $N_{\mathrm{par}}(S,a)$, and $R(S,a)$ denote payload bits, serialized parameter count, and routing bits. For model cost λ in bits per parameter, define

$$J^\star(\lambda) = \min_{S,a:\ \mathrm{UCB} \le \tau} \left[P(S,a) + R(S,a) + \lambda N_{\mathrm{par}}(S,a)\right].$$

Theorem 2 (Cost path and break-even)**.** *$J^\star(\lambda)$ is concave and piecewise linear. With equal per-model cost, the parameter count of its optimizer is non-increasing in λ. As $\lambda \to \infty$ the optimizer attains the feasible plan of smallest serialized parameter count, so the plateau is $K^\star = 1$ exactly when a certified global model attains that minimum. A shared plan beats local fallback exactly when*

$$P_{\mathrm{local}} - P_{\mathrm{shared}} > \lambda(N_{\mathrm{par,shared}} - N_{\mathrm{par,local}}) + (R_{\mathrm{shared}} - R_{\mathrm{local}}).$$

In our real-data settings no global predictor is certified on every admitted cell, so the plateau is not reached. Section 4 reports the empirically observed cost path.

The certificate changes which assignments are feasible. Let $r(n_c, M, \delta)$ be the simultaneous finite-sample certificate radius and $R_c(h)$ the population risk of action h in cell c. The next result bounds the price of this restriction in terms of the certificate radius r.

Proposition 2 (Certificate preservation)**.** *On the certificate event, which holds with probability at least $1 - \delta$, every calibration-certified assignment satisfies $R_c(a(c)) \leq \tau$ for all cells. For a fixed objective J, the population feasible set at $\tau - 2r$ is contained in the certified feasible set, which is contained in the population feasible set at τ. Therefore $J^\star(\tau) \leq J^\star(\text{certified}) \leq J^\star(\tau - 2r)$.*

The certificate constrains contract risk. Section 4.1 measures calibration-to-test payload generalization directly and finds selection optimism of at most 0.0002 bits/sample. The protocol is fit $\rightarrow$ tune (fix H, M) $\rightarrow$ calibration (certify and assign) $\rightarrow$ test. Every admitted cell has a certified local action. The general subset problem is NP-hard by reduction from Set Cover (Appendix A.4). The uniform-cost coverage surrogate is monotone submodular, so greedy maximization has a $1 - 1/e$ guarantee (Krause & Golovin, 2014); that guarantee does not cover the full objective. Exact enumeration is polynomial when $K = O(\log m)$. The experiments use $K \leq 16$. The next proposition states the calibration guarantee available when the portfolio contains a comparison family such as the baseline partitions.

Proposition 3 (Baseline containment)**.** *If the candidate portfolio contains every model induced by a baseline partition family Π_{base}, then*

$$J_{\text{cal}}(\text{CSDS}) \leq \min_{\pi \in \Pi_{\text{base}}} J_{\text{cal}}(\pi).$$

A planted-structure analysis (Appendix A.4.2) gives the sharing boundary: once the between-group gap exceeds twice the certificate radius, the certified compatibility graph recovers the planted groups and the optimum switches between full sharing and local fallback at an explicit bit threshold.

4 Experiments

The results answer five questions. **RQ1:** does certified selection reduce deployment cost? **RQ2:** does certification control contract risk? **RQ3:** what limits certified sharing? **RQ4:** which layer produces the gain? **RQ5:** can the failure budget be reallocated across candidates?

Setup. All methods use the same fit/tune/calibration/test split $(48/12/20/20)$ and the same fitted predictors. A cell is admitted only if its local predictor satisfies $\text{UCB}_c(\text{local}(c)) \leq \tau = 0.40$. The headline predictor is CART with maximum depth five, 16 bits per serialized parameter, up to 128 contract cells, and certificate level $\delta = 0.05$ over 30 seeds. Payload is negative log-likelihood code length, and model bits use the serialized fitted parameters. The all-local and clustered baselines fit one predictor per cell and one per cluster; the learned router uses KMeans cell signatures and lets calibration select both $K \leq 16$ and a certified cluster action or local fallback by the same deployment objective, so every baseline uses the same contract, certificate and cost accounting. The four Covertype contracts share a base dataset. Served cells per setting are in Appendix A.3.1.

Statistical analysis. Aggregate differences use 10,000 two-level paired bootstrap replicates that resample settings and then seeds, with equal weight per setting; we report the 2.5th and 97.5th percentiles. Test data are used only for reporting the final results.

4.1 Does Certified Selection Reduce Deployment Cost?

CSDS lowers deployment cost against all-local deployment in every one of the 150 paired runs, by 4.97 to 5.52 bits per test sample, and against the calibration-tuned router in every Covertype run and 25 of 30 on Electricity, by $+0.80$ to $+3.70$ bits (see Table 1).

Where the savings come from. CSDS routes only 12.3%–16.0% of served Covertype cells to local fallback against 61.8%–70.9% for the $K \leq 16$ router, and 23.7% against 36.9% on Electricity, so the gain is carried by the stored local models that certified sharing removes rather than by a lower negative log-likelihood. Selection optimism is bounded: the selected certified subset exceeds the test-optimal certified subset by at most 0.0002 ± 0.0007 bits/sample (Appendix Table 10).

Setting	All-local	Router	CSDS (Ours)	Gain vs local	Gain vs router	Fallback
Covertype W1	7.638 ± 0.055	5.797 ± 0.314	**2.334 ± 0.217**	**+5.304 ± 0.215 (30/30)**	**+3.463 ± 0.363 (30/30)**	13.3%
Covertype W3	7.296 ± 0.058	5.008 ± 0.273	**2.322 ± 0.181**	**+4.974 ± 0.185 (30/30)**	**+2.686 ± 0.363 (30/30)**	14.5%
W1 × Soil30	7.890 ± 0.065	6.071 ± 0.298	**2.368 ± 0.198**	**+5.522 ± 0.188 (30/30)**	**+3.702 ± 0.367 (30/30)**	12.3%
W3 × Soil30	7.562 ± 0.061	5.519 ± 0.272	**2.532 ± 0.189**	**+5.030 ± 0.197 (30/30)**	**+2.987 ± 0.373 (30/30)**	16.0%
Covertype average	7.597 ± 0.221	5.599 ± 0.487	**2.389 ± 0.212**	**+5.208 ± 0.294 (120/120)**	**+3.210 ± 0.538 (120/120)**	14.0%
Electricity demand	9.317 ± 0.237	4.813 ± 0.547	**4.016 ± 0.617**	**+5.301 ± 0.621 (30/30)**	**+0.796 ± 0.829 (25/30)**	23.7%

Table 1: **Deployment cost under hard contracts.** Five 30-seed settings, in bits/sample. Best and second-best costs are bold and underlined. Lower cost is better. Gains are baseline minus CSDS. Router selects $K \leq 16$ on calibration by the same deployment objective; full $K \leq 8$ results are in Appendix Table 8. Fallback is the CSDS local-fallback fraction.

Coverage is part of the contract. The served fraction is a reported deployment quantity: 120.7 to 122.9 of 128 Covertype cells receive a certified predictor, against 32.4 of 90.9 on Electricity. The rest are referred, so Table 1 prices the served stream. Charging every contract cell preserves the ordering and most of the margin, CSDS leading all-local by 4.74 to 5.26 bits on Covertype and 2.54 on Electricity, and pricing referral at κ bits per sample keeps the Covertype ordering at every κ tested while making all-local cheapest on Electricity only beyond $\kappa \approx 16.1$ (Appendix Table 9).

4.2 Does Certification Control Contract Risk?

Under the headline Hoeffding certificate no served cell in any of the five settings shows a held-out contract violation, local fallback included. These counts are a consistency check on the per-deployment guarantee, not an estimate of its coverage. Removing the certificate on two additional contracts produces violations immediately, 13.03 and 1.80 per seed (Appendix Figure 3).

Certificate families differ in their finite-sample behavior. Under six certificate families, Hoeffding, one-sided Hoeffding and empirical Bernstein (Maurer & Pontil, 2009) produce no held-out violation anywhere, while Clopper–Pearson and Bentkus (Bates et al., 2021) each show one violated cell across 30 Covertype W1 seeds. Conformal risk control (Angelopoulos et al., 2024) is different in kind: it controls an expected risk rather than a per-cell bound, certifies no shared action in our contracts, and produces six held-out violations on Electricity, so we report it as a negative control and exclude it from the certified deployment interface (Appendix Table 7).

4.3 What Limits Certified Sharing?

Shared-action capacity is the fraction of admitted cells with at least one certified candidate shared action. Empirical calibration risk admits every shared action, so capacity is 1.000 in all five settings and any shortfall is the price of a finite-sample guarantee, and two deployment choices set that price.

Calibration support dominates. At full support the Hoeffding certificate certifies 85.0%–88.8% of served Covertype cells and 81.8% on Electricity, costing 11.2 to 18.2 points; at a quarter of the calibration data it certifies 43.4%–53.3% on Covertype and at a tenth only 2.2%–14.7%. The bottleneck is not the contract or the heterogeneity but the calibration evidence each cell carries, a quantity that the contract's own cell width fixes (see Figure 7).

The certificate family is worth points. Clopper–Pearson recovers 4.3 to 6.3 points on Covertype and 6.1 on Electricity over Hoeffding at the same δ, while empirical Bernstein gives away 21 to 27; restricting the pool to the tune-selected columns costs a further few points (Appendix Table 12). Certification is priced by evidence and by the bound, and choosing the bound is free at deployment time, since it changes no fitted parameter and consumes no additional calibration data.

4.4 Which Layer Produces the Gain?

Certification, not search, is the scarce resource. Growing the candidate pool moves the margin against the router monotonically from -1.469 bits at a baseline-compatible pool of two columns through -0.460 at four to $+2.021$ at eight and $+2.647$ at full budget, turning a loss in all 150 runs into a win in all 150 (Appendix Table 22). Removing the certificate instead produces immediate violations

Control	Δ	95% CI	W/T/L
All-local deployment	$+5.226$	$[5.046, 5.412]$	150/0/0
Clustered router, $K \leq 16$	$+2.727$	$[1.701, 3.470]$	145/0/5
Best fixed-K, same candidates	0.000	$[0.000, 0.000]$	0/150/0

Table 2: **Paired controls across the five settings** (bits/sample; 30 seeds each). Intervals are two-level paired bootstrap percentiles. Charging every contract cell for every method instead gives $+4.481$ $[3.474, 5.112]$ and $+2.527$ $[1.408, 3.312]$, with the same win counts; per-dataset values for both accountings, and the referral-cost sweep, are listed in Appendix A.3.4.

(Section 4.2). The selection layer is not where the value sits: against a matched-candidate control, a best calibration-selected fixed-K policy drawing on the identical certified pool, CSDS ties in all 150 runs at every support level and on two further scarce contracts, so the gain is carried by the certified action set, not the combinatorics (Table 2; Appendix Table 19).

Although subset choice is NP-hard in general, the certified cost path is flat near its minimum on these contracts, so the engineering effort belongs in the certificate and the candidate pool. Against a certified MoE-style router on the same 30 seeds, CSDS leads on the four Covertype contracts by 3.10 to 4.02 bits in every seed and on Electricity by 0.18 in 19 of 30 (Appendix Table 20).

4.5 Can the Failure Budget Be Reallocated?

The certificate layer supplies a second decision variable, and unlike subset search it is not idle at full support. Because the tune split is independent of calibration, the failure probability can be split unevenly across candidates as long as the parts sum to δ; we give weight four to the four tune-ranked leaders and weight one to the rest, fixing the rule before any calibration label is read. On the four Covertype contracts at full support this raises shared-action capacity by 7.3 to 9.9 points and lowers cost by 0.618 ± 0.185 to 0.786 ± 0.194 bits/sample in every seed, with larger effects at scarce support (Appendix Figure 5). A rule ablation over $k \in \{2, 4, 8\}$ stays within 0.033 bits/sample of the rule chosen on test. Reallocation moves observed risk as well as cost: the weighted arm showed two violated cells at full support and two more at reduced support across the $4 \times 6 \times 30$ grid, against none for uniform allocation, so all other results use uniform allocation (Appendix Table 18; Figures 6 and 4). The certified interface carries over to deeper CART and MLP portfolios and to a four-domain LoRA adapter deployment on a frozen backbone (Appendix A.3.7).

5 Conclusion

CSDS makes certified sharing coverage a deployment optimization variable. Cell-separability lets $m(K+1)$ certified cell–action pairs cover every deployment plan at once, and a cost-aware portfolio over that certified set beats all-local deployment and a tuned router with no observed violation. Matched controls identify the certificate, not subset search, as the source of the saving.

AI Use Statement

Generative AI tools were used in two ways: to aid and polish the writing of this manuscript, and for retrieval and discovery when locating related work. The authors designed the method and the experimental protocol, implemented and executed all experiments, verified every reported measurement against the underlying artifacts, and checked the cited literature. No generative AI tool was used to generate data, experimental measurements, or results. The authors made all final scientific and editorial decisions and take full responsibility for the contents of this paper.

Ethics Statement

This work uses public benchmark datasets and does not involve human subjects, personal data, or deployed decision-making. The risk certificates provide statistical guarantees for the specified contract cells and do not replace domain-specific validation.

REPRODUCIBILITY STATEMENT

The experiments use deterministic fit/tune/calibration/test splits, public tabular datasets, fixed random seeds, and finite-sample certificates computed only from calibration data. The appendix defines the contract construction, candidate portfolios, solvers, certificate computation, and evaluation protocol. The anonymized code in the supplementary material contains the candidate generator, exact and greedy solvers, aggregation scripts, and commands for the main tables and figures. All reported test metrics come from held-out data. The neural deployment uses IMDB, Yelp, Amazon, and Rotten Tomatoes with rank-8 LoRA adapters on `q_lin` and `v_lin`.

REFERENCES

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, pp. 55198–55218, 2024.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Léonard Blier and Yann Ollivier. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31, 2018.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 624–633, 2014.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010.

António Farinhas, Chrysoula Zerva, Dennis Ulmer, and André F T Martins. Non-exchangeable conformal risk control. In *International Conference on Learning Representations*, pp. 50952–50966, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19586–19597, 2020.

Alex Graves. Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, 24, 2011.

Peter D Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *International Conference on Learning Representations*, 2016.

Úrsula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pp. 1939–1948. PMLR, 2018.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2018.

Laurent Jacob, Francis Bach, and Jean-Philippe Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Angela H Jiang, Daniel L-K Wong, Christopher Canel, Lilia Tang, Ishan Misra, Michael Kaminsky, Michael A Kozuch, Padmanabhan Pillai, David G Andersen, and Gregory R Ganger. Mainstream: Dynamic Stem-Sharing for Multi-Tenant video processing. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pp. 29–42, 2018.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pp. 2564–2572. PMLR, 2018.

Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pp. 71–104. Cambridge University Press, 2014.

Bracha Laufer-Goldshtein, Adam Fisch, Regina Barzilay, and Tommi Jaakkola. Risk-controlling model selection via guided Bayesian optimization. *arXiv preprint arXiv:2312.01692*, 2023a.

Bracha Laufer-Goldshtein, Adam Fisch, Regina Barzilay, and Tommi S Jaakkola. Efficiently controlling multiple risks with Pareto testing. In *International Conference on Learning Representations*, 2023b.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in Neural Information Processing Systems*, 30, 2017.

Ivan Rubachev, Nikolay Kartashev, Yury Gorishniy, and Artem Babenko. TabReD: Analyzing pitfalls and filling the gaps in tabular deep learning benchmarks. In *International Conference on Learning Representations*, pp. 35166–35202, 2025.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter-efficient multi-task model fusion with partial linearization. In *International Conference on Learning Representations*, pp. 21148–21178, 2024.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel J Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, second edition, 2022.

Haowen Wang, Tao Sun, Congyun Jin, Yingbo Wang, Yibo Fan, Yunqi Xu, Yuliang Du, and Cong Fan. Customizable combination of parameter-efficient modules for multi-task learning. In *International Conference on Learning Representations*, pp. 2504–2525, 2024.

Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022.

A.1 Assumptions and Notation

The notation defines the objects used by every certificate and selection argument.

- $C = \{1, \dots, m\}$ is the set of contract cells.
- $H = \{1, \dots, K\}$ is the finite set of shared predictor candidates.
- $S \subseteq H$ is the selected shared set.
- $a(c) \in S \cup \{\text{local}(c)\}$ is the action assigned to cell c.
- $L_c(h)$ is the calibration payload cost, $L_c^{\text{test}}(h)$ the test payload cost, and $b(h)$ the serialized model cost.
- $A(S, a)$ is the deployed assignment-code length. The implementation uses a fixed-width routing index, $A(S, a) = \sum_{c \in C} \lceil \log_2(|S| + 1) \rceil$.
- $\text{UCB}_c(h)$ is a finite-sample upper confidence bound on the risk of action h in cell c.
- ϵ_{cell} is the protected-field resolution used to form numeric contract cells.

The local fallback is a certified action for each admitted cell, following the admission rule in Section 3. Candidate predictors are fit without test data. Calibration and test samples are disjoint. Within each cell the calibration losses are assumed independent and identically distributed and independent of the fitted candidates, which holds because candidates are fit on the training split and the portfolio is fixed on the tune split. The certificate makes no assumption across cells; the union bound handles the simultaneous statement. Table 3 places CSDS against the closest families.

A.2 Deployment Portfolio Overview

Family	Sharing degree	Assignment	Risk control	Deployment cost
MTL	architecture	fixed	task	parameters
MoE	experts	learned	aggregate	routing
Clustered	clusters	hard	fallback	models
MDL	compression	–	–	model
CSDS	subset S	per-cell	per-cell	all

Table 3: Positioning. CSDS makes certified sharing coverage and cell assignment the optimization variables under a per-cell risk contract and a deployment bit budget.

A.3 Supplementary Analyses

A.3.1 Experimental Protocol Details

Headline runs use a fit/tune/calibration/test split of $48/12/20/20$. The candidate family H, its cardinality K, and therefore the multiplicity $M = m(K + 1)$ are fixed on the fit and tune splits, before any calibration label is used. The $+1$ carries the per-cell local admission test, so the m local certificates and the mK shared certificates draw on one budget δ. Calibration data are used only to certify cell–action pairs and to select the deployed portfolio and assignment within the certified set. Test labels are revealed only after the assignment is fixed. The predictor is CART with maximum depth five, serialized at 16 bits per parameter. CART model bits use the fitted node and leaf counts rather than the maximum capacity of its depth. The risk budget is $\tau = 0.40$ unless stated otherwise, and the certificate level is $\delta = 0.05$.

Because the four Covertype contracts share a base dataset, the paired runs measure stability across seeds and contract constructions rather than across independent environments; Electricity is the one setting drawn from a different source.

Serving a cell only when its own local predictor is certified restricts the policy class, and we measure what that costs by counting, per seed, the referred cells on which some candidate in $\mathcal{H}$ is nonetheless

certified. Over 30 seeds these are 43 of 3840 contract cells on Covertype W1, 70 of 3840 on W3, 66 and 63 of 3840 on the Soil30 variants, and 208 of 2728 on Electricity, that is 1.1 to 1.8 percent of the contract on Covertype and 7.6 percent on Electricity, where serving them would raise coverage from 35.6 to about 43 percent. The restriction is mild where local predictors are easy to certify and material where they are not, so Electricity is where admitting cells on shared evidence alone would gain most. It costs coverage, not validity: every served cell keeps a certified fallback, which is what makes the per-cell minimization feasible for every subset.

Each contract cell is the product of a protected indicator and a binned protected numeric field, the bin width being ϵ_{cell}. The binned field carries essentially all of the cell resolution and is removed from the feature matrix seen by every predictor, for CSDS and for every baseline, so no predictor can recover which cell a sample belongs to. The indicator remains an ordinary feature, available identically to every method. Table 4 reports admitted cells and calibration support per setting, and Table 5 the composition of the certified cells.

A cell is admitted only if its local predictor is certified, $\mathrm{UCB}_c(\mathrm{local}(c)) \leq \tau$. Rejected cells are routed to a manual or reject action outside the serialized stream.

Setting	Test samples	Admitted cells	Excluded cells	Cal/cell
Covertype W1	74,124	122.9	5.1	580.6
W1 × Soil30	70,264	121.6	6.4	550.8
Covertype W3	74,275	121.4	6.6	584.0
W3 × Soil30	70,330	120.7	7.3	552.7
Electricity demand	8,818	32.4	58.5	131.6

Table 4: Admitted cells and calibration support. Excluded cells fail the local UCB admission test.

Setting	$K^\star$	Assigned to certified shared action (%)	Forced local (%)	Chosen local (%)
Covertype W1	7.60 ± 0.93	86.7	12.5	0.8
Covertype W3	7.23 ± 0.94	85.5	13.3	1.2
W1 × Soil30	8.23 ± 0.82	87.7	11.2	1.0
W3 × Soil30	7.67 ± 1.03	84.0	15.0	1.0
Electricity demand	2.27 ± 0.78	76.3	18.2	5.6

Table 5: Certified-cell composition, pooled over 30 seeds. Assigned to a certified shared action: the selected subset contains a certified action for the cell. Forced local: no candidate shared action is certified for the cell. Chosen local: at least one candidate shared action is certified for the cell, but no certified shared action enters the selected subset.

Protected fields define contract cells. Categorical fields define one cell per observed category, while numeric fields use a predeclared resolution. Product contracts use intersections of the protected fields. Cells with insufficient training support are excluded, and the retained cells are capped at 128. The definitions depend only on protected fields and the training split.

The local, global, and clustered baselines fit one predictor per cell, one predictor for all cells, and one predictor per cluster. The learned router (Shazeer et al., 2017) uses KMeans clusters of cell signatures, one model per cluster, and calibration selects $K \leq 16$ and a certified cluster action or local fallback by the same deployment objective. It uses the same certificate and fallback rule as CSDS. Covertype uses Wilderness Area and Soil Type contracts with Elevation as the protected numeric field. Electricity uses the day/demand contract. Table 6 collects the predictor and candidate diagnostics.

A.3.2 Predictor and Candidate Diagnostics

Check	Evidence	Result
Candidate recovery	4-group mixture	100/100
Predictor family	MLP + Clopper	5/10 certified
Scalability	10,000 cells	0.038 s, 80.7 MB
Calibration objective	Five settings	145/150

Table 6: Predictor and candidate diagnostics.

A.3.3 Certificate and Accounting Sensitivity

The five 30-seed headline suites contain 15,567 served cell-seed observations with no observed violation. The certificate changes the selected assignment while preserving the contract.

The certificate ablation isolates the cost of enforcing the contract. On Covertype W2, removing the upper confidence bound produces 13.03 mean held-out violations per seed, while the UCB selector produces none. On Electricity price, the corresponding no-certificate variant produces 1.80 mean violations.

	Local admission		Shared capacity		
Bound	Covertype	Electricity	Covertype	Electricity	Viol.
Empirical Bernstein	0.677	0.008	0.631	0.333	0
Hoeffding, two-sided	0.950	0.356	0.870	0.818	0
Hoeffding, one-sided	0.957	0.382	0.881	0.823	0
Clopper–Pearson	**0.980**	**0.531**	**0.922**	**0.879**	1
Bentkus	**0.980**	**0.531**	**0.922**	**0.879**	1
CRC quantile	1.000	0.200	0.000	0.000	6

Table 7: Risk-control alternatives under the same calibration protocol, 30 seeds per setting. Covertype columns are means over the four Covertype contracts. Viol. counts held-out contract violations summed over all five settings. CRC controls an expected risk rather than a per-cell bound: it certifies no shared action and is the only family that violates the contract, so we report it as a negative control.

Setting	Router bits ($K \leq 8$)	Gain ($K \leq 8$)	Router bits ($K \leq 16$)	Gain ($K \leq 16$)
Covertype W1	6.273	+3.939	5.797	+3.463
W1 $\times$ Soil30	6.529	+4.161	6.071	+3.702
Covertype W3	5.475	+3.153	5.008	+2.686
W3 $\times$ Soil30	5.857	+3.325	5.519	+2.987
Electricity demand	4.615	+0.599	4.813	+0.796

Table 8: Router protocol comparison in bits/sample. The main paper uses the $K \leq 16$ protocol throughout. The $K \leq 16$ search has a larger candidate-action universe, so its simultaneous union-bound radius is wider; the wider search can therefore have higher test bits on Electricity despite considering more clusters.

A.3.4 Coverage-Adjusted Deployment Cost

Table 1 prices the certified stream, that is, the cells that pass local admission. Table 9 adds the accounting in which excluded cells are restored and served by their own local predictors, so that every contract cell is charged. The ordering of the three methods is unchanged in all five settings. Because every method serves the excluded cells with the same local predictors, the full-contract margin is the certified-stream margin scaled by the admitted sample fraction: it retains 95% of the Covertype gain and 48% of the Electricity gain, where roughly two thirds of the contract cells never enter the certified stream. The same two-level paired bootstrap applied to this accounting gives $+4.481$ $[3.474, 5.112]$ against all-local and $+2.527$ $[1.408, 3.312]$ against the router over all five settings, and

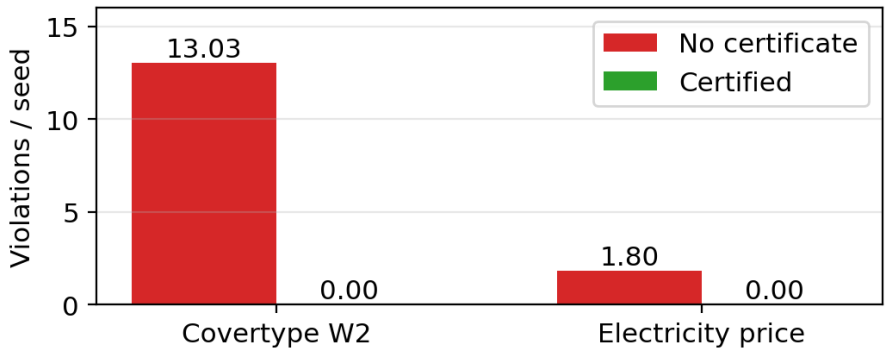

Figure 3: No-certificate and empirical selection allow held-out contract violations. The UCB certificate removes them.

$+4.966$ $[4.749, 5.199]$ and $+3.061$ $[2.678, 3.450]$ over the four Covertype settings. The win counts match the certified-stream comparison exactly, at 150/0/0 and 145/0/5 over five settings and 120/0/0 over four, so the conclusion does not depend on which of the two accountings is taken as primary.

Setting	Method	Certified stream	All contract cells
Covertype W1	all-local	7.638 ± 0.055	7.646 ± 0.054
	router $K \leq 16$	5.797 ± 0.314	5.875 ± 0.312
	CSDS	$\mathbf{2.334 \pm 0.217}$	$\mathbf{2.546 \pm 0.220}$
W1 $\times$ Soil30	all-local	7.890 ± 0.065	7.898 ± 0.065
	router $K \leq 16$	6.071 ± 0.298	6.164 ± 0.280
	CSDS	$\mathbf{2.368 \pm 0.198}$	$\mathbf{2.639 \pm 0.177}$
Covertype W3	all-local	7.296 ± 0.058	7.320 ± 0.060
	router $K \leq 16$	5.008 ± 0.273	5.139 ± 0.264
	CSDS	$\mathbf{2.322 \pm 0.181}$	$\mathbf{2.579 \pm 0.179}$
W3 $\times$ Soil30	all-local	7.562 ± 0.061	7.585 ± 0.056
	router $K \leq 16$	5.519 ± 0.272	5.650 ± 0.260
	CSDS	$\mathbf{2.532 \pm 0.189}$	$\mathbf{2.821 \pm 0.191}$
Electricity demand	all-local	9.317 ± 0.237	10.304 ± 0.114
	router $K \leq 16$	4.813 ± 0.547	8.156 ± 0.208
	CSDS	$\mathbf{4.016 \pm 0.617}$	$\mathbf{7.764 \pm 0.405}$

Table 9: Deployment cost in bits/sample under two accountings, mean $\pm$ standard deviation over 30 seeds. Certified stream charges only admitted cells. All contract cells restores the excluded cells and serves them with their local predictors, which are not certified. Charging excluded cells a manual cost of κ bits/sample instead preserves the ordering CSDS $<$ router $<$ all-local at every tested κ on all four Covertype contracts; on Electricity all-local becomes cheapest beyond $\kappa \approx 16.1$ bits/sample.

A.3.5 Selection Optimism

The exact solver minimizes a calibration objective over 2^{12} subsets, so the selected plan could be optimistic on held-out data. Table 10 compares it with the subset that minimizes the test objective over the same certified family. That oracle is diagnostic only and is not deployable.

A.3.6 Shared-Action Capacity Denominators

Shared-action capacity counts cells with at least one certified shared action. The main text uses admitted cells as the denominator, which matches the deployed stream. Table 11 also reports all contract cells, where the Electricity value is much lower because most of its cells fail local admission before any shared action is considered.

Setting	Optimism gap	Selected subset equals oracle
Covertype W1	0.0000 ± 0.0000	100.0%
W1 × Soil30	0.0002 ± 0.0007	90.0%
Covertype W3	0.0000 ± 0.0002	96.7%
W3 × Soil30	0.0000 ± 0.0001	96.7%
Electricity demand	0.0000 ± 0.0002	96.7%

Table 10: Optimism of calibration-based subset selection over 30 seeds. The gap is the test objective of the selected subset minus the test objective of the subset that minimizes the test objective over the same certified family, in bits/sample. The oracle is diagnostic only.

Setting	Bound	All contract cells	Admitted cells
Covertype W1	Empirical	$100.0\% \pm 0.0\%$	$100.0\% \pm 0.0\%$
	Hoeffding	$84.0\% \pm 3.2\%$	$87.5\% \pm 3.1\%$
	Clopper–Pearson	$90.6\% \pm 2.2\%$	$92.1\% \pm 2.2\%$
W1 × Soil30	Empirical	$100.0\% \pm 0.0\%$	$100.0\% \pm 0.0\%$
	Hoeffding	$84.3\% \pm 2.8\%$	$88.8\% \pm 3.0\%$
	Clopper–Pearson	$91.3\% \pm 2.3\%$	$93.0\% \pm 2.7\%$
Covertype W3	Empirical	$100.0\% \pm 0.0\%$	$100.0\% \pm 0.0\%$
	Hoeffding	$82.2\% \pm 3.0\%$	$86.7\% \pm 2.8\%$
	Clopper–Pearson	$90.3\% \pm 2.3\%$	$92.4\% \pm 2.4\%$
W3 × Soil30	Empirical	$100.0\% \pm 0.0\%$	$100.0\% \pm 0.0\%$
	Hoeffding	$80.1\% \pm 3.7\%$	$85.0\% \pm 3.8\%$
	Clopper–Pearson	$89.2\% \pm 2.6\%$	$91.3\% \pm 2.5\%$
Electricity demand	Empirical	$98.4\% \pm 1.2\%$	$100.0\% \pm 0.0\%$
	Hoeffding	$29.2\% \pm 3.4\%$	$81.8\% \pm 6.6\%$
	Clopper–Pearson	$46.7\% \pm 3.4\%$	$87.9\% \pm 5.1\%$

Table 11: Shared-action capacity under two denominators, mean ± standard deviation over 30 seeds at full calibration support.

Certifying only the tune-selected columns, rather than every generated candidate, is what the deployed portfolio pays for being small. Table 12 isolates that cost at full calibration support under the Hoeffding certificate.

Setting	Pre-tune pool	Deployed pool	Cost (points)
Covertype W1	94.8%	87.5%	7.3
W1 × Soil30	95.5%	88.8%	6.8
Covertype W3	94.3%	86.7%	7.5
W3 × Soil30	93.5%	85.0%	8.6
Electricity demand	84.0%	81.8%	2.2

Table 12: Shared-action capacity on admitted cells before and after tune-time candidate selection, Hoeffding certificate at full calibration support. The pre-tune pool is every generated candidate; the deployed pool is the tune-selected portfolio that the certificate multiplicity is computed from.

A.3.7 Neural Deployment Volume

Beyond CART. Deeper CART and MLP portfolios stay positive against the router in every seed with no violation (Table 17), and a four-domain LoRA deployment on frozen DistilBERT backbones reproduces the cost path of Theorem 2: at small volume the optimum is the single global adapter, which CSDS selects, and a certified shared portfolio takes over once payload amortizes the extra adapter bits (Appendix A.3.7).

The static comparison uses 1,200 test samples per domain. At $\tau = 0.5$, CSDS selects one shared adapter in all three seeds. Table 14 reports the full 1,200-sample accounting. The horizon sweep is in

Horizon N	$K^\star$ distribution	Shared cells	Gain (M bits)	Gain/sample	Wins	Viol.
10M	1 (10/10)	8.0	0.000	0.000	0/10	0
30M	1 (4/10), 2 (6/10)	8.0	2.403 ± 3.334	0.080	5/10	0
50M	2 (10/10)	8.0	8.507 ± 6.541	0.170	9/10	0
70M	1 (1/10), 2 (8/10), 3 (1/10)	7.0	**10.156 ± 6.862**	**0.145**	**10/10**	0
100M	1 (8/10), 2 (2/10)	4.6	8.526 ± 8.670	0.085	9/10	0

Table 13: Break-even horizon analysis. Gain is global minus CSDS in serialized bits. $K^\star$ is the number of shared adapters, not the number of covered cells. Cells without a certified shared adapter use local fallback. Shared cells is the mean number of cells covered by shared adapters out of eight.

the main paper (Table 13). In the horizon sweep, $K^\star$ counts shared adapters rather than covered cells. A $K^\star = 1$ plan can be a global adapter over all eight cells or one partial adapter with local fallback; all $K^\star = 1$ plans at the 100M-sample horizon use the latter. Unlike the CART contracts, the neural local action is a domain-level adapter: four local adapters cover the eight domain-length cells, and each serialized adapter is counted once.

Method	Bits (M)	Risk	Viol.
Local adapters	47.335	0.094	0
Global adapter	**11.835**	0.111	0
Cluster router	11.836	0.145	0
MoE-lite router	55.322	0.095	0
CSDS	**11.835**	**0.111**	0

Table 14: Static neural adapter deployment over four domains and three seeds. Local adapters are four domain adapters; bits are serialized adapter bits over 1,200 test samples per domain.

A.3.8 Risk-Budget Sensitivity

Figure 4 varies the contract risk budget on Electricity demand while holding the predictor, candidate portfolio, and splits fixed. At $\tau = 0.4$, every seed admits a certified portfolio and the selector uses multiple shared predictors on average. The sweep therefore treats τ as a contract-level input rather than a tuned performance parameter.

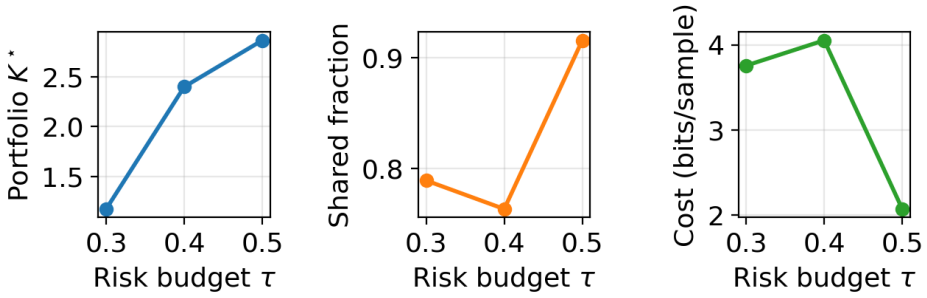

Figure 4: Risk-budget sensitivity on Electricity demand, 10 seeds.

A.3.9 Additive-Cost Sensitivity

The theory requires an additive cost rather than serialized bits in particular. Rescaling model and routing costs by 0.5 and 2.0 preserves positive gain over the learned router in $10/10$ Covertype W1 seeds, $10/10$ Covertype W1×Soil30 seeds, and $26/30$ IPUMS employment-by-age seeds. Serialized bits remain the headline unit. This tests the same selection layer under alternative relative costs.

A.3.10 SELECTION SCALABILITY

Table 15 benchmarks the exact selection layer as the number of contract cells grows to ten thousand. With eight candidates, exact selection runs in 0.070 seconds and 80.7 MB at 10,000 cells. Increasing the candidate count shows the expected exponential growth and motivates the greedy variants used by the implementation.

Cells	Candidates	Exact time (s)	Peak memory (MB)	Greedy gap (%)
128	8	0.017	1.0	59.8
1,024	8	0.022	8.3	17.4
10,000	8	0.070	80.7	3.9
1,024	10	0.094	33.0	10.8
1,024	12	0.358	132.1	5.9

Table 15: Selection-layer scalability.

A.3.11 CANDIDATE COMPLETENESS

Table 16 evaluates whether the candidate portfolio recovers a planted sharing structure. We generate four latent groups with four cells per group, add Gaussian signature noise, and compare the best candidate partition with the planted partition. At low and moderate noise, the candidate family recovers the planted sharing structure exactly in almost all trials. The experiment isolates candidate-set coverage. The guarantee of Theorem 1 applies to the generated portfolio.

Signature noise	Trials	Exact recovery	Mean relative gap
0.05	100	100/100	0.0000
0.10	100	100/100	0.0000
0.20	100	100/100	0.0000
0.35	100	98/100	0.0005

Table 16: Candidate-completeness check against a planted sharing structure.

Predictor	Setting	Gain ($K \leq 16$)	Gain ($K \leq 8$)	Wins	Viol.
CART depth 5	W1	+3.463	+3.939	30/30	0
	W1×Soil30	+3.702	+4.161	30/30	0
CART depth 10	W1	**+12.703**	**+14.209**	**30/30**	0
	W1×Soil30	**+13.215**	**+15.106**	**30/30**	0
MLP	W1	+2.242	+2.368	10/10	0
	W1×Soil30	+2.851	+2.911	10/10	0

Table 17: Predictor-family control. CART rows use 30 seeds under the headline protocol; MLP rows use 10 seeds at 32 cells. Gains are bits per test sample against the clustered router under both router protocols.

The selection delta, that is the exact solver minus the best fixed-K policy on the same certified candidates, is identically zero in all 30 seeds at every one of the six calibration-support levels, for every candidate budget in $\{2, 4, 8, 12\}$ and in all five settings. We therefore report the tie as a single number rather than a curve.

A.3.12 FAILURE-BUDGET ALLOCATION

Table 18 reports the tune-weighted allocation against uniform Bonferroni on the four Covertype contracts, 30 seeds per cell of the grid. Capacity gain is weighted minus uniform shared-action capacity; cost gain is uniform minus weighted bits/sample, so positive is better. Seeds in which neither arm admits a feasible shared action have no defined cost and are excluded from the cost column, with their count given.

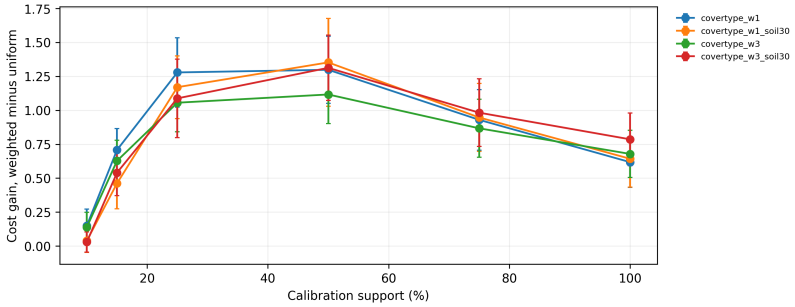

Figure 5: **The certificate layer still pays at full support.** Tune-weighted against uniform allocation of the same δ, over calibration support. Bands are standard errors over 30 seeds; the dashed line is the mean over the four Covertype contracts.

Support	Capacity gain	Wins	Cost gain	Wins	Weighted violations
Covertype W1					
10%	0.371 ± 0.497	13/30	0.205 ± 0.120	22/22	0
15%	0.497 ± 0.174	30/30	0.708 ± 0.156	30/30	0
25%	0.399 ± 0.081	30/30	1.278 ± 0.251	30/30	0
50%	0.213 ± 0.042	30/30	1.298 ± 0.242	30/30	0
75%	0.130 ± 0.034	30/30	0.930 ± 0.219	30/30	1
100%	0.078 ± 0.025	30/30	0.618 ± 0.182	30/30	1
W1 × Soil30					
25%	0.393 ± 0.082	30/30	1.170 ± 0.226	30/30	0
100%	0.073 ± 0.027	30/30	0.643 ± 0.205	30/30	0
Covertype W3					
25%	0.341 ± 0.059	30/30	1.056 ± 0.210	30/30	0
100%	0.089 ± 0.027	30/30	0.679 ± 0.170	30/30	1
W3 × Soil30					
25%	0.380 ± 0.091	30/30	1.088 ± 0.285	30/30	0
100%	0.099 ± 0.034	30/30	0.786 ± 0.191	30/30	0

Table 18: Tune-weighted failure-budget allocation against uniform Bonferroni, mean $\pm$ standard deviation over 30 seeds. Wins counts seeds with a positive gain, out of the seeds with a defined value. The uniform arm produced no held-out violation anywhere in the grid. Intermediate support levels for the last three settings follow the same pattern as Covertype W1 and are in the released per-seed file. The main-text 0.033 bits/sample ablation gap is the maximum over all six support levels, not a full-support-only result; the largest gap occurs at 50% support.

A.3.13 Additional Scarce-Support Settings

Table 19 adds two settings with less calibration support per cell than the headline contracts: a Covertype product contract at a finer elevation resolution, which yields 512 cells, and an IPUMS employment-by-age contract. Both gain over all-local deployment in every seed with no observed violation, and in both the exact solver ties the fixed-K policy in every seed. Local admission is also much stricter here, so these settings price a small certified stream.

A.3.14 Predictor-Family Scope

Certified outcomes hold across predictor families. On Covertype W1 a CART portfolio certifies in all 30 seeds, selects $K^\star = 7.07$ shared predictors on average, and gains $+1.141$ bits per test sample over the $K \leq 8$ router. A neural MLP portfolio on the CIFAR domain contract certifies in 5 of 10 seeds under the Clopper–Pearson certificate, which is tighter than Hoeffding at those per-cell sample sizes, and gains $+0.104$ bits averaged over the certified seeds. Every certified run has zero test-risk violations.

	Covertype W1 × Soil30, $\epsilon = 1$	IPUMS empstat × age
Contract cells	512	29.4
Calibration samples per cell	149.8	48.9
Gain over all-local (bits/sample)	19.650 (30/30)	6.741 (30/30)
Gain over fixed-K (bits/sample)	0.000 (0/30)	0.000 (0/30)
Violations	0	0

Table 19: Two additional contracts with reduced per-cell calibration support, 30 seeds each. Percentages and gains are means over seeds; parentheses give the number of seeds with a positive gain.

A.3.15 Baseline-Family Containment

Across the five headline settings and 30 seeds per setting, the CSDS calibration objective is no larger than the calibration-tuned router objective in 145 of 150 runs. Proposition 3 states a sufficient condition on the portfolio; the deployed portfolio is capped at twelve columns, so the 145/150 figure is an empirical calibration comparison rather than an instance of the proposition.

A.3.16 Fixed-K and Certified MoE Baselines

Table 20 compares the adaptive portfolio with the best fixed-K policy and a certified MoE-style router. Fixed-K uses the same candidate set and assignment rule but restricts the number of shared predictors. The certified MoE router trains cluster experts and a calibration-time gate, then applies the same risk certificate and fallback. Fixed-K and certified MoE both use 30 seeds. All runs have zero test-risk violations.

Setting	**Gain vs fixed-K**	Fixed-K wins	**Gain vs MoE**	MoE wins
Covertype W1	0.000	30 ties	**+3.878**	**30/30**
W1 × Soil30	0.000	30 ties	**+4.018**	**30/30**
Covertype W3	0.000	30 ties	**+3.100**	**30/30**
W3 × Soil30	0.000	30 ties	**+3.283**	**30/30**
Electricity demand	0.000	30 ties	**+0.178**	19/30

Table 20: Fixed-K and certified MoE controls in bits/test sample. The fixed-K gain is zero in all settings with 30 ties per setting; the certified MoE control uses the same 30 seeds and the same fitted experts.

A.3.17 Candidate-Reference Gap

Table 21 compares the candidate portfolio with an exhaustive partition-induced reference on small contract instances. The reference enumerates every nonempty cell subset as a candidate training pool and every set partition as an action set, using the same certificate and fixed-width routing. The objective gap measures coverage within the enumerated reference, which enumerates 3,487 to 4,140 partition-induced actions on 8-cell instances and is far larger than any deployable portfolio. The heuristic family is better on three of the five contracts and behind by at most 0.19 bits/sample on the other two, so the candidate-constrained limitation costs a bounded and small amount here rather than an unknown one.

Setting	Cells	Seeds	**CSDS advantage**	Positive
Covertype W1	8	10	−0.128	1/10
W1 × Soil30	8	10	−0.187	0/10
Covertype W3	8	10	+0.042	4/10
W3 × Soil30	8	10	+0.181	7/10
Electricity demand	8	10	**+0.307**	**7/10**

Table 21: Candidate-reference objective gap in bits/test sample.

Candidate generation uses KMeans with ten initializations for local signatures, feature profiles, and target profiles. Cluster counts range from two to the candidate budget. Baseline partitions are generated for one through eight clusters. The hierarchy is average linkage. The headline configuration uses at most twelve candidates and up to four tune-selected extra candidate predictors. A candidate predictor is retained only when its tune payload is finite and its empirical risk satisfies the contract. Extra candidates maximize marginal payload savings minus model and routing cost.

A.3.18 Portfolio and Candidate-Budget Control

The portfolio control separates the contribution of the candidate family from the selection layer. With only baseline-compatible candidates, the mean gain against the learned router is $+0.697$ bits and four of five seeds are positive. The baseline-only portfolio retains eight baseline columns and is not a complete instance of Π_{base}. The full portfolio increases the gain to $+0.872$ bits, makes all five seeds positive, and reduces the mean selected portfolio size from 4.2 to 3.4 actions. Every run retains zero test-risk violations.

The candidate-budget control varies only the number of candidate predictors. The mean gain against the $K \leq 16$ router rises from -1.469 with a baseline-compatible pool of two columns, through -0.460 at four and $+2.021$ at eight, to $+2.647$ at the full budget, while the selected portfolio grows from 1.78 to 6.60 actions. Certified coverage, not the search over subsets, is what turns the comparison positive.

Portfolio	Seeds	Gain vs router	Positive	Mean $K^\star$	Viol.
Baseline-only (2)	150	-1.469	0/150	1.78	0
Candidates 4	150	-0.460	30/150	3.07	0
Candidates 8	150	+2.021	**150/150**	5.35	0
Full portfolio (12)	150	**+2.647**	**150/150**	6.60	0

Table 22: **Candidate coverage is what produces the gain.** Requested candidate budget against the $K \leq 16$ router, five settings $\times$ 30 seeds, bits per test sample. With a baseline-compatible pool of two columns CSDS is worse than the router in every run; the gain turns positive only once the portfolio is large enough for the certificate to admit shared actions. Each seed is capped at its tune-selected candidate count, so the requested budget of twelve corresponds to an effective mean of 10.1.

A.3.19 Hyperparameter Response

Figure 6 reports the two hyperparameters that move the deployment decision. Deployment cost falls monotonically with the candidate budget in all five settings, which is the coverage effect of Section 4.4 seen from the supply side: a larger candidate family gives the certificate more shared actions to admit. The certificate family moves capacity far more than any selection choice, and the ordering empirical $>$ Clopper–Pearson $>$ Hoeffding is stable across settings. The risk budget τ is swept separately in Appendix A.3.8, where only Electricity has a matching cached protocol.

A.4 Theory and Proofs

Exact Selection. For a fixed shared set S, the shared-model cost $\sum_{h \in S} b(h)$ is constant, and the fixed-width routing code gives

$$A(S, a) = \sum_{c \in C} \lceil \log_2(|S| + 1) \rceil.$$

This term is constant as well. The remaining objective is

$$\sum_c \left(L_c(a(c)) + b_{\text{local}}(c) \mathbf{1}[a(c) = \text{local}(c)] \right).$$

Each term depends only on $a(c)$. Minimizing each term subject to $\text{UCB}_c(a(c)) \leq \tau$ therefore minimizes the sum. There are $O(m|S|)$ candidate actions to evaluate.

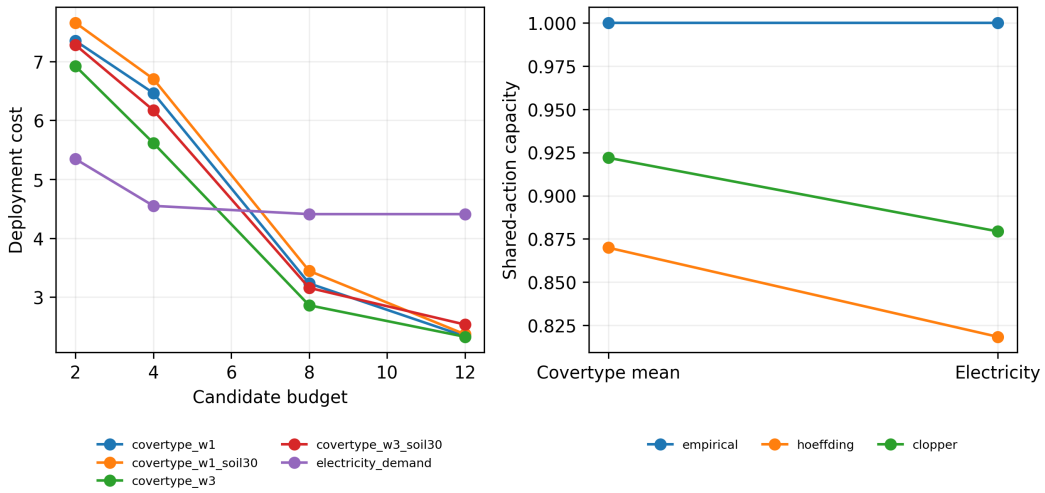

Figure 6: **Hyperparameter response.** (a) Deployment cost against the candidate budget $|H|$, from the candidate-budget sweep protocol. (b) Shared-action capacity by certificate family at full calibration support, on the admitted-cell denominator. Error bars are standard deviations over 30 seeds.

For finite H, each feasible subset S is evaluated by the exact per-cell assignment above. The finite union of these optimal fixed-set objectives contains the global optimum over the candidate portfolio, so the minimizing subset is optimal over H.

General Subset Hardness. We reduce Set Cover to CSDS. Let $(U, \mathcal{F})$ be a Set Cover instance. Each element $u \in U$ becomes a contract cell. Each set $F \in \mathcal{F}$ becomes a shared predictor. Predictor F is feasible on cell u exactly when $u \in F$. Set payload gains to zero and per-model cost to one; the routing term $m\lceil \log_2(|S|+1) \rceil$ of the deployed objective is left untouched, since it depends on S only through $|S|$ and is nondecreasing in it, so a minimizer of the deployment objective is still a minimum-cardinality feasible cover. Local fallback is feasible for every cell and has cost larger than any candidate cover.

Any solution cheaper than local fallback selects a collection of shared predictors whose feasible cells cover all elements. Its cost equals the number of selected sets. Thus a polynomial-time exact solver for CSDS would solve Set Cover. The reduction instantiates the objective with a local fallback cost larger than any candidate cover; local fallback remains a certified action for every admitted cell.

Approximation Lower Bound. The reduction in the previous section preserves the objective value of a Set Cover solution. Therefore a polynomial-time approximation algorithm for CSDS with factor $(1-\varepsilon)\ln m$ would imply the same approximation for Set Cover. The result follows from the $(1-\varepsilon)\ln|U|$ inapproximability of Set Cover for every $\varepsilon > 0$ unless P=NP (Dinur & Steurer, 2014).

Uniform-Cost Coverage Surrogate. Let $f(T)$ be the number of cells covered by a shared set T under equal model costs and a cardinality budget. Coverage is monotone: adding a predictor cannot remove a covered cell. For any $T_1 \subseteq T_2$ and predictor $h \notin T_2$, the marginal coverage of h is non-increasing:

$$f(T_1 \cup \{h\}) - f(T_1) \geq f(T_2 \cup \{h\}) - f(T_2).$$

The set of cells uncovered by T_1 contains the set of cells uncovered by T_2, so the new coverage contributed by h can only shrink. Hence f is monotone submodular. Greedy maximization obtains the standard $(1-1/e)$ guarantee for this coverage surrogate (Krause & Golovin, 2014). It does not imply an approximation guarantee for the full objective because payload, model, assignment, and local-fallback costs are not represented by cell coverage.

Model-Cost Break-Even. Let C_{local} be the local fallback cost and C_{shared} the shared cost. Let N_{local} and N_{shared} be the corresponding serialized parameter counts, and let λ be the model cost in

bits per parameter. Write

$$C_{\text{local}} = P_{\text{local}} + \lambda N_{\text{local}} + A_{\text{local}}, \qquad C_{\text{shared}} = P_{\text{shared}} + \lambda N_{\text{shared}} + A_{\text{shared}},$$

where P denotes payload cost and A is assignment code length. Define the payload reduction $G = P_{\text{local}} - P_{\text{shared}}$. Sharing is optimal exactly when $C_{\text{shared}} < C_{\text{local}}$, which is equivalent to

$$G > \lambda(N_{\text{shared}} - N_{\text{local}}) + A_{\text{shared}} - A_{\text{local}}.$$

For a shared family S, the shared model term is $\lambda \sum_{h \in S} N_{\text{par}}(h)$ and the same argument applies.

Pairwise Separation. There are g pairs of cells. In each pair, the two cells have opposite Bayes-optimal label functions, so no single predictor is feasible for both cells. Therefore a single global predictor is infeasible.

The local solution pays one model per cell:

$$2g(B + nc_{\text{local}}).$$

The pair-shared solution pays one model per pair:

$$g(B + 2nc_{\text{shared}}).$$

Subtracting the shared cost from the local cost gives

$$2g(B + nc_{\text{local}}) - g(B + 2nc_{\text{shared}}) = g(B + 2n(c_{\text{local}} - c_{\text{shared}})).$$

The pair-shared solution is strictly cheaper exactly when $B > 2n(c_{\text{shared}} - c_{\text{local}})$.

Baseline Containment. Every baseline partition $\pi \in \Pi_{\text{base}}$ induces an assignment of cells to shared actions. Under the hypothesis that every model used by such an assignment is present in the CSDS candidate portfolio, the CSDS feasible set therefore contains every baseline assignment. Since CSDS minimizes the calibration objective over its feasible set,

$$J_{\text{cal}}(\text{CSDS}) \leq \min_{\pi \in \Pi_{\text{base}}} J_{\text{cal}}(\pi).$$

Tune-then-Certify Selection. Portfolio selection depends only on the tune split, and candidate fitting is independent of the calibration and test splits. Conditioned on the tune-selected portfolio, the tested family is $\{(c, h) : c \in C,\ h \in H\} \cup \{(c, \text{local}(c)) : c \in C\}$, so the number of tested cell-action pairs is $M = m(K + 1)$; the local admission tests are included because $\text{local}(c)$ is an action of the deployed assignment. Hoeffding's inequality applied at failure level δ/M and a union bound over the M pairs make all certificates simultaneously valid with probability at least $1 - \delta$. On this event, every calibration-feasible action has population risk at most τ. Therefore any calibration-measurable assignment from the portfolio satisfies the per-cell risk constraints.

Certified Family Portfolio. Let $\mathcal{F}$ be the family portfolio, and let δ_f be the certificate level of family f. For a fixed deterministic selection rule, each cell receives an action from at least one selected family. If all selected family certificates are correct, the resulting action assignment is feasible. A union bound over family failures gives total failure probability at most

$$\sum_{f \in \mathcal{F}} \delta_f \leq \delta.$$

Hoeffding Certificate. For a cell-action pair with n bounded losses in $[0, 1]$, Hoeffding's inequality gives

$$\Pr\left(|\hat{R} - R| \geq \sqrt{\frac{\log(2M/\delta)}{2n}}\right) \leq \frac{\delta}{M}.$$

A union bound over M tested cell-action pairs makes all certificates simultaneously valid with probability at least $1 - \delta$.

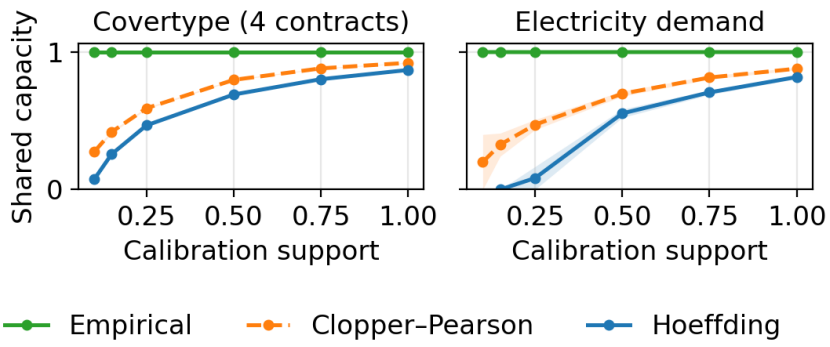

Figure 7: **The certification price is a function of calibration support.** Shared-action capacity on admitted cells, averaged over the four Covertype contracts and shown separately for Electricity. Bands are standard errors over 30 seeds.

A.4.1 Cost and Certificate Results

Cost path and plateau. Each feasible subset contributes the line $A + \lambda N_{\mathrm{par}}$. The pointwise minimum of a finite family of lines is concave and piecewise linear. If $\lambda_1 < \lambda_2$, optimality at both points gives

$$(\lambda_2 - \lambda_1)(N_{\mathrm{par},2} - N_{\mathrm{par},1}) \le 0,$$

so $N_{\mathrm{par},2} \le N_{\mathrm{par},1}$. As $\lambda \to \infty$, the leading term is the serialized parameter count. The optimizer therefore selects the feasible plan of smallest serialized parameter count. That plan is the certified global model exactly when its parameter count is no larger than every other feasible plan's, in which case the plateau is $K^\star = 1$; otherwise the plateau is whichever feasible plan attains the minimum, which may be all-local when the individual local models are small.

Certificate price and lower bound. If $R_c(h) \le \tau - 2r$ and $\mathrm{UCB} \le R + 2r$ on the certificate event, then $\mathrm{UCB} \le \tau - 2r + 2r = \tau$. Thus every action that is feasible at the stricter population budget is certified at τ, while every certified action is population-feasible at τ on the certificate event. For a fixed objective, the resulting feasible-set inclusion gives $J^\star(\tau) \le J^\star(\mathrm{certified}) \le J^\star(\tau - 2r)$. This comparison isolates the effect of the certificate on the feasible set; calibration-to-population error in the payload objective is measured separately by selection optimism. The lower bound follows by rearranging the two-sided Hoeffding radius $r = \sqrt{\log(2M/\delta)/(2n_c)}$.

A.4.2 Planted Sharing Boundary

The planted model splits the m cells into G equal groups, each with its own predictor. Write ρ_0 for that predictor's population risk on a cell of its own group, so within-group risks lie in $[\rho_0, \rho_0 + \varepsilon]$ and cross-group risks are at least $\rho_0 + \Delta$. Two cells are joined in the *certified compatibility graph* when one predictor is certified on both. Let n be the samples per cell, r the certificate radius of Proposition 2, B the serialized bits of one shared model, and $c_{\mathrm{shared}}, c_{\mathrm{local}}$ the per-sample payload bits under shared and local predictors.

Theorem 3 (Planted sharing boundary)**.** *If $\Delta - \varepsilon > 2r$ and $\tau \in (\rho_0 + 2r, \rho_0 + \Delta - 2r)$, then the certified compatibility graph equals the planted group structure with probability at least $1 - \delta$. In the symmetric planted construction, the optimum uses all G shared groups when*

$$B > \frac{m}{m - G}\, n\, (c_{\mathrm{shared}} - c_{\mathrm{local}}).$$

and uses local fallback otherwise.

This boundary is verified on the synthetic planted family in Figure 8.

Planted phase boundary. When $\Delta - \varepsilon > 2r$, separate group centers remain farther than twice the certificate radius, while within-group deviations do not create cross-group certified edges. The

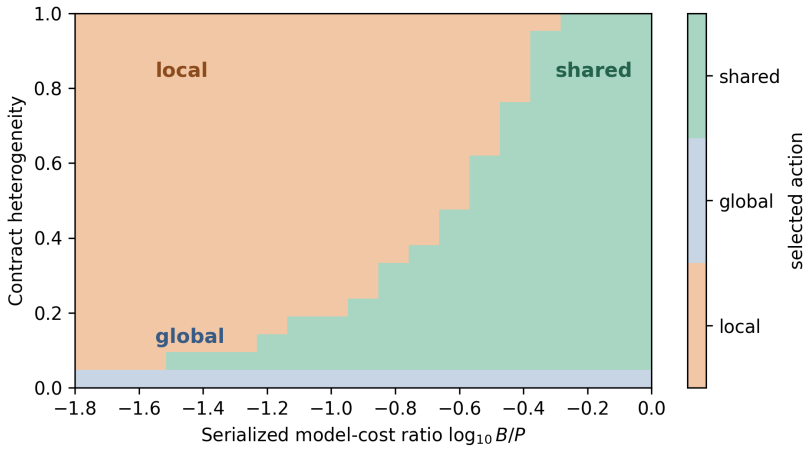

Figure 8: Deployment sharing map in a synthetic contract family.

certified compatibility graph is therefore the planted group graph. Because the planted groups are symmetric, a partial sharing solution is dominated by either selecting all G group models or using local fallback. Comparing G shared models against m local models gives

$$GB + mnc_{\text{shared}} < m(B + nc_{\text{local}}),$$

which rearranges to the break-even condition $B > \frac{m}{m-G} n(c_{\text{shared}} - c_{\text{local}})$. The optimum is $K^\star = G$ when this condition holds and $K^\star = 0$ otherwise.