WHEN DOES INPUT ADAPTATION NEED A SELECTOR? THE ROLE OF ROUTING-CAPABLE BASELINES

Anonymous authors
Paper under double-blind review

ABSTRACT

Input adaptation gives a frozen predictor several possible corrections. A learned selector can choose one for each example. Its value, however, depends on what a single candidate can already do. We study this dependence by adding routing candidates built from the same fitted maps. This enlargement preserves the pointwise oracle for fixed row-wise maps. It strengthens the best complete candidate and cannot increase selector headroom. An exact value decomposition locates the remaining losses in a learned tree. The controlled study makes the effect concrete. With 200 development examples, the original selector gains 9.6 accuracy points over the frozen predictor. A routed candidate gains 12.4 points using the same maps. Across 36 datasets, cross-fitted selectors add little to enlarged libraries with three frozen networks. Split learning and TabICLv2 retain positive increments, showing that the outcome depends on the learning setup. Natural-shift sensitivity checks separate adaptation gains from inactive input pathways. Finally, we distinguish testing an overall improvement from testing a selector upgrade. The resulting evaluation compares selection with a baseline that already supports simple routing. This separates routing gains from residual selector value.

1 INTRODUCTION

A selector is useful only relative to the alternatives it can choose. This matters when adapting a frozen predictor to a shifted target distribution. One can fit several input corrections on a small labeled sample (Chhabra et al., 2019). A learned selector then chooses a correction for each example. The apparent benefit can be large. Yet that benefit may reflect a weak comparison library rather than a need to learn another decision rule.

The key distinction is between a single *map* and a single *candidate procedure*. Suppose two maps have been fitted to two clusters. A complete candidate can send each cluster through its own map. It is one candidate, although its computation depends on the input. Comparing a selector only with globally applied maps overlooks this option. The selector can receive credit for routing that was already available from the fitted partition and corrections.

We therefore strengthen the comparison library before evaluating selection. We retain the original maps and add cluster-conditional and piecewise candidates. No map is refitted for this enlargement. A cluster-conditional candidate changes inputs in one cluster and keeps identity elsewhere. A piecewise candidate uses the matching correction in both clusters. The default and the learned selector then operate on the same enlarged library. Their difference measures the value left for selection after this simple form of routing is available.

The controlled experiment illustrates why this comparison matters. With identifiable subpopulations and 200 development examples, a tree selector gains 9.6 percentage points (pp) over the frozen predictor. A routed candidate built from the same maps gains 12.4 pp. After that candidate enters the library, the selector gains 11.7 pp. Its residual increment is negative. The example changes the question from whether routing helps to whether *learning another routing rule* helps beyond a strong complete candidate.

Our analysis uses the gap between the pointwise best candidate and the best complete candidate. We call this gap selector headroom. Routing among fixed maps can improve the latter without changing the former. Headroom therefore cannot increase. A learned tree also loses value through

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.

its partition and its leaf actions. An exact accounting identity separates these losses from the shortfall of its selected default. This distinction matters because a weak default can inflate the measured selector increment.

The experiments test this argument at several levels. Controlled shifts expose conditional benefits and permit population accounting. Constructed shifts on 36 OpenML datasets test different frozen models and learning procedures. Natural shifts across five survey tasks test the pattern beyond synthetic perturbations. Cross-fitted selectors add little to routing-capable libraries with three networks. Split learning gives positive increments, and TabICLv2 adds 0.80 pp. These results identify when the conclusion changes instead of treating a small network average as a universal property.

Deployment introduces a separate comparison. A procedure can beat the untouched predictor without beating an available default. A selector upgrade must be tested against that default. We state the corresponding paired tests and their assumptions. The paper thus contributes an evaluation argument with three parts. It makes routing capacity explicit in the baseline, accounts for residual selector value, and matches each deployment claim to its reference. These parts use established selection and testing tools. Together, they make gains easier to attribute and comparisons easier to reproduce.

2 Setup and candidate construction

Let f be a frozen classifier and P the target distribution. Candidate T_k defines the complete procedure $h_k(x) = \arg\max_y f_y(T_k(x))$. Identity $T_0 = \mathrm{id}$ gives the untouched predictor. A candidate may include a feature-dependent routing rule. For context-based models, it may also transform context features while keeping their labels fixed. The row-wise theory applies when each prediction depends only on its query and a fixed context.

We divide the target sample into development, guard, and test data. Development data fit maps, choose a default, and train the selector. Guard data support the deployment decision. Test data measure final performance. Candidate and selector fitting do not use guard or test labels. The frozen-model constraint keeps the model parameters and downstream decision rule fixed. It permits input changes and the resulting changes in predictions.

Candidate libraries. The real-data network library $\mathcal{A}$ has six candidates. They are identity, mean matching, moment matching, translation, featurewise affine transformation, and low-rank affine transformation. Library $\mathcal{B}$ adds four cluster-fitted maps. It fits moment matching and low-rank maps within each of two k-means clusters, then applies each map globally. Library $\mathcal{B}'$ adds four cluster-conditional candidates and two piecewise candidates. It therefore contains 16 complete procedures. The partition and fitted maps stay fixed during enlargement. The synthetic study uses fewer map families, as specified in Appendix B.

Map fitting and default selection. Learned maps start at identity. They minimize a Brier objective for at most 80 Adam steps. An inner validation quarter selects the stopping point. Columns with at most two distinct source values are protected. The selected default $\hat{g}$ maximizes positive out-of-fold accuracy gain over identity. Identity is retained if no candidate has positive gain. The main tie rule uses Brier improvement and then simplicity. The cached routing-library analysis uses simplicity alone because its caches lack Brier scores. We identify that configuration when reporting its results.

Paired learning targets. Let c_{ik} be candidate k's correctness on a development example whose label was excluded from that candidate's fit. The selector uses

$$a_{ik} = c_{ik} - c_{i\hat{g}}, \qquad a_{i\hat{g}} = 0. \tag{1}$$

Both terms use the same default index. A fold-specific reference can otherwise make a candidate look beneficial even when it is worse than the final default. The paired target removes that mismatch. Choosing the default on the same matrix can still introduce selection bias.

Learning constructions. Five-fold cross-fitting supplies development correctness values. Maps are then refitted on all development data. The selector thus trains on fold-wise maps and deploys with refitted ones. In the split construction, half D_1 fits maps and selects the default. The other half D_2

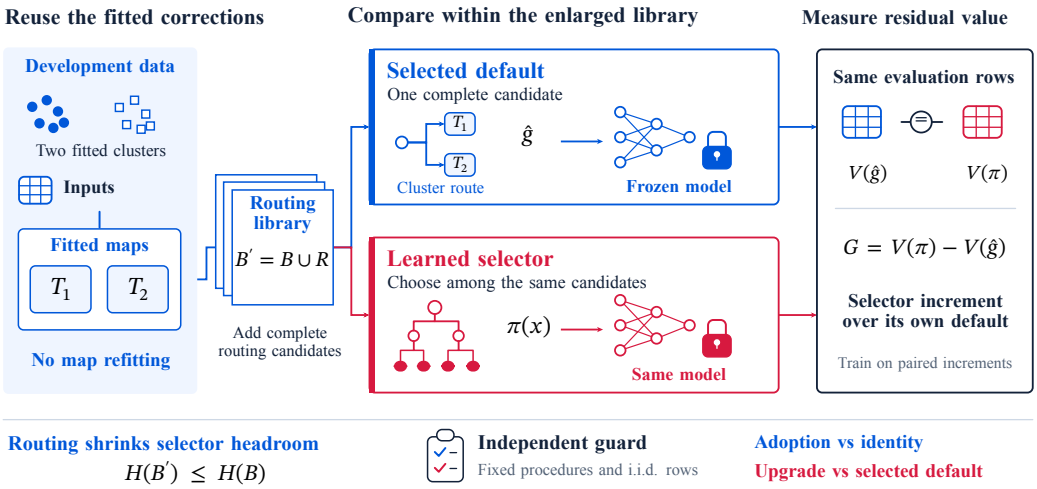

Figure 1: **Evaluate selection after routing is available to the baseline.** Shared maps form an enlarged library. The blue branch selects one complete candidate, shown here as a cluster route. The red branch learns a selector over the same library. Both use the same frozen model. Paired evaluation measures the residual increment. The guard comparisons address distinct deployment claims. Icons illustrate mechanisms, not measured data.

trains the selector. Clustering still uses all development features, and the split is label-stratified. We treat it as a different empirical construction, rather than an independent validation design.

The main selector is a depth-two greedy tree. Each leaf contains at least $\max(30, n/10)$ examples. Split selection maximizes unpenalized paired increments. A leaf switches only when its best mean increment exceeds 1.645 estimated standard errors. This penalty is a training heuristic. The independent guard test supplies the deployment comparison. Figure 1 summarizes these roles.

3 Selector Headroom and Value Accounting

All quantities in this section are conditional on the fitted library and refer to the target population. Define the conditional benefit of candidate k over the untouched predictor by

$$b_k(x) = \mathbb{E}[\ell(h_0(X), Y) - \ell(h_k(X), Y) \mid X = x], \qquad V(\pi) = \mathbb{E}[b_{\pi(X)}(X)], \tag{2}$$

where ℓ is zero–one loss and π reads features only. Write

$$V_{\text{point}} = \mathbb{E}[\max_k b_k(X)], \qquad V_{\text{best}} = \max_k \mathbb{E}[b_k(X)], \qquad H(\mathcal{T}) = V_{\text{point}} - V_{\text{best}}. \tag{3}$$

A selector attaining the pointwise maximum reads conditional benefits, not realized labels. It is distinct from a hindsight oracle that chooses a candidate after observing each outcome. The difference in Equation (3) is nonnegative. A heterogeneous shift can still have $H = 0$. One candidate may be optimal almost everywhere. Conversely, an imperfect library can have $H > 0$ under a homogeneous shift.

Proposition 1 (Routing enlargement)**.** *Fix a finite library of row-wise candidates $\mathcal{T}$ and a feature-space partition. Let $\mathcal{T}' \supseteq \mathcal{T}$ add only candidates that apply a member of $\mathcal{T}$ within each cell. Then*

$$V_{\text{point}}(\mathcal{T}') = V_{\text{point}}(\mathcal{T}), \quad V_{\text{best}}(\mathcal{T}') \geq V_{\text{best}}(\mathcal{T}), \quad H(\mathcal{T}') \leq H(\mathcal{T}). \tag{4}$$

If a cell-wise selector's assignment is itself included in $\mathcal{T}'$, its value is no larger than $V_{\text{best}}(\mathcal{T}')$.

Proof. At any x, each added candidate agrees with an existing candidate. It cannot improve the pointwise maximum, while inclusion of the original library prevents that maximum from decreasing. Maximizing a complete-candidate value over a larger set cannot reduce V_{best}. Subtracting gives the headroom inequality. The last claim follows because the corresponding complete procedure belongs to the enlarged library. □

Refitting or adding maps can change both oracle values.

Value accounting for a learned tree. For the terminal partition $\mathcal{L}$ of a fitted selector, let

$$V_{\text{leaf}} = \sum_{L \in \mathcal{L}} P(X \in L) \max_k \mathbb{E}[b_k(X) \mid X \in L]. \tag{5}$$

Assigning each learned leaf its best population candidate defines a diagnostic oracle. Put

$$R_{\text{partition}} = V_{\text{point}} - V_{\text{leaf}}, \qquad R_{\text{action}} = V_{\text{leaf}} - V(\pi), \qquad R_{\text{default}} = V_{\text{best}} - V(\hat{g}). \tag{6}$$

All three terms are nonnegative. Adding and subtracting the intermediate values gives

$$\boxed{V(\pi) - V(\hat{g}) = H - R_{\text{partition}} - R_{\text{action}} + R_{\text{default}}} \leq H + R_{\text{default}}. \tag{7}$$

The default shortfall enters with a positive sign. An imperfect default makes an increment easier to obtain. The partition term describes the *learned* partition, not the irreducible limitation of all depth-two trees. The action term combines estimation, regularization, and refitting effects. The identity locates losses without assigning causal shares to these mechanisms.

The controlled generator permits Monte Carlo evaluation of these quantities. On real data, we instead compare held-out paired outcomes. Deployment still requires the intended guard comparison.

4 Testing deployment claims

Deployment requires a specified reference. Let A be a fitted procedure and R its reference. Define $p_+ = P(R \text{ wrong}, A \text{ right})$ and $p_- = P(R \text{ right}, A \text{ wrong})$. The accuracy difference is $p_+ - p_-$. On independent, identically distributed guard examples, count these outcomes as n_+ and n_-. Set $m = n_+ + n_-$. The exact one-sided paired test is

$$p_{\text{guard}} = \Pr\{\text{Bin}(m, 1/2) \geq n_+\}. \tag{8}$$

For $m = 0$, set $p_{\text{guard}} = 1$. Reject $p_+ \leq p_-$ when $p_{\text{guard}} \leq \alpha$. This conditional binomial form uses the paired comparison associated with McNemar's test (McNemar, 1947). Appendix A gives the validity argument.

Testing against identity addresses overall accuracy improvement. Testing a selector against $\hat{g}$ addresses a selector upgrade. These are different claims. If the default is already deployed, the upgrade test directly answers whether to replace it. If both claims are required, the testing plan must cover both. One development-selected procedure can receive one guard test against identity. A pre-specified family can instead use Holm correction (Holm, 1979).

A fixed-sequence rule first tests the default against identity. It tests the selector against the default only after rejection (Westfall & Krishen, 2001). This order can lose power when the default adds little but selection adds much. If the default is identity, the first comparison has no discordances and cannot reject. Bypassing this comparison would define a different rule. The original nested-adoption tables require an implementation check on this point. We record it in Appendix C and do not use those rates to support the main empirical claim.

The guard argument conditions on procedures fixed before evaluation. Each paired outcome must depend only on its own guard row. The reported TabICLv2 batch probe and the online BETA implementation do not establish this premise. We report their performance without an exact row-wise guard guarantee. The test controls repeated-sample false rejection for its chosen accuracy comparison. Calibration and subgroup outcomes require their own evaluation.

5 Experiments

We test three questions. Does routing change the attribution of adaptation gains? What accounts for the value left to a learned selector? Does that pattern persist across models and natural shifts? All numerical results come from the supplied manuscript and supplement. Appendix D maps each comparison to its source. The plots reorganize those results without adding new experimental runs.

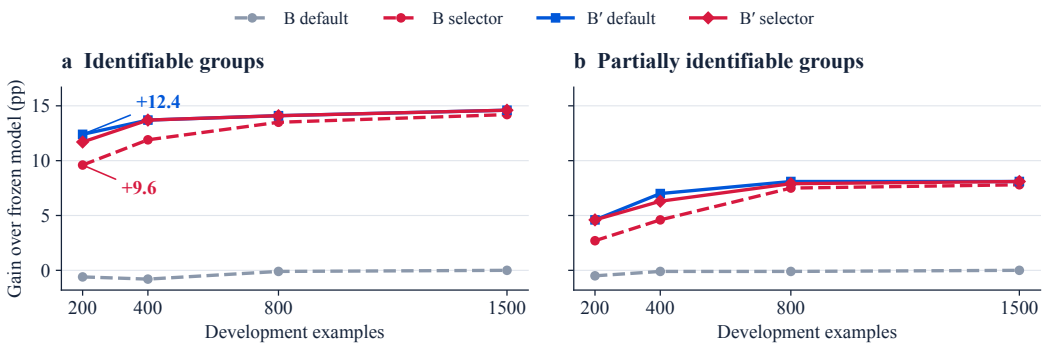

Figure 2: **A routed baseline captures much of the apparent selector gain.** Accuracy gains over the frozen model use the same fitted maps within each comparison. Blue shows the routed default. Red shows selectors, with dashed lines for the original library. Each point is the mean of ten repetitions from Table S8. At 200 examples, the routed default gains 12.4 pp and the original selector gains 9.6 pp.

5.1 Controlled shifts expose the effect of the candidate library

The generator draws four-dimensional Gaussian features with logistic labels. A latent group rotates the features by $+50°$ or $-50°$. An observed site feature identifies the group with Bayes accuracy $\Phi(\delta)$. Values $\delta \in \{3, 1, 0\}$ give identifiable, partial, and non-identifiable groups. The frozen model ignores the site feature. Null and homogeneous shifts provide controls. A fixed library contains identity and the two inverse rotations. Learned libraries fit their maps on development examples.

Population accuracy is available through the generator's conditional label probabilities. Each value averages 10^5 independent feature draws. The learned-library results average ten fitting repetitions. The fixed-library results average 40. This design separates the effect of fitted maps from test-label noise. Variation across fitted repetitions remains distinct from Monte Carlo integration error.

Library enlargement changes the comparison. With 200 examples and identifiable groups, $\mathcal{B}$ has headroom $H = 13.6$ pp. Its default has gain -0.6 pp, while its selector gains 9.6 pp. Routing enlargement reduces headroom to 1.2 pp. The $\mathcal{B}'$ default gains 12.4 pp and its selector gains 11.7 pp. Thus the residual selector gain changes from $+10.2$ to -0.7 pp. The map fits are shared. The change comes from which complete procedures the baseline is allowed to select.

Figure 2 also shows how sample size changes the result. In the identifiable setting, the original selector rises from 9.6 to 14.2 pp between 200 and 1,500 examples. The routed default rises from 12.4 to 14.6 pp. The gap narrows as the tree has more data to learn the routing rule. With partial group information, the routed default gains 4.6 pp at 200 examples. The original selector gains 2.7 pp. Both reach about 8 pp at the largest sample size. The largest benefit of the stronger baseline appears where learning a separate selector is most data-limited.

The controls identify what creates headroom. Without cluster-fitted maps, library $\mathcal{A}$ leaves only 2.2 pp of headroom in the identifiable setting. Its selector gains 0.1 pp. A selector needs useful alternatives before it can exploit group structure. Under homogeneous shift, the fixed library contains the exact inverse and has $H = 0$. Under non-identifiable groups, no fixed candidate improves on identity pointwise. The fitted tree nonetheless loses 1.2 pp at 200 examples. These controls separate the existence of shift, the availability of useful corrections, and the ability to infer which correction to use.

5.2 Value accounting locates the remaining tree losses

The diagnostic study evaluates 60 fitted runs at 200, 400, and 800 examples. It holds the learned leaves fixed and evaluates their actions under the known conditional distribution. The diagnostic runs are separate from those in Figure 2. Figure 3 displays their headroom and two loss terms.

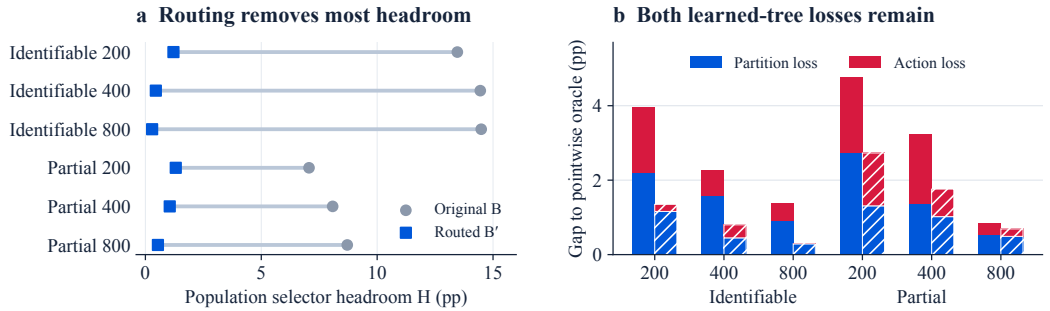

Figure 3: **Routing reduces headroom before tree estimation becomes the main issue.** Left, the enlarged library has much less headroom in every diagnostic setting. Right, partition and action losses sum to the gap from the pointwise oracle. Each pair shows $\mathcal{B}$ and hatched $\mathcal{B}'$. Values are means from the separate fits in Table S11. Bars show the components directly and do not imply causal attribution.

In $\mathcal{B}$, the total gap to the pointwise oracle ranges from 0.85 to 4.75 pp. Both loss terms are positive in all six settings. Partition loss is larger in five. For identifiable groups at 200 examples, partition and action losses are 2.21 and 1.75 pp. At 800 examples they fall to 0.91 and 0.47 pp. The tree approaches the available conditional opportunity as the learning sample grows.

Library $\mathcal{B}'$ changes the scale of the problem. Its headroom ranges from 0.29 to 1.31 pp. Its mean selector increment ranges from -0.59 to 0.00 pp. At 800 identifiable examples, its 0.29 pp headroom equals the partition loss and its action loss is zero. A more accurate leaf-action estimator alone cannot recover that gap for the learned partition. With partial information at 200 examples, the action loss is larger. The accounting distinguishes these settings even when both produce small residual gains.

The decomposition also explains why default quality must be reported. In the identifiable $\mathcal{B}$ setting at 200 examples, the selected default falls 0.99 pp below the best complete candidate. This shortfall increases the observed selector increment by the same amount. A large increment can therefore combine useful selection with an avoidably weak reference. Comparing only selector minus selected default conceals that distinction.

A targeted change aligns the tree's split score with its penalized leaf action. Independent evaluation does not support it as a general repair. In $\mathcal{B}$, alignment changes the pooled increment by -0.72 pp under greedy search. Its 95% interval is $[-1.02, -0.40]$. Finite exact search gives -0.39 pp. In $\mathcal{B}'$, greedy alignment reduces the average deficit to -0.06 pp but gives no positive mean increment. Improving the training objective is therefore insufficient to establish a better selector.

5.3 Model and learning construction determine the residual gain

The OpenML study uses 36 numerical-feature datasets. Each has a no-shift control, selection bias, an affine perturbation, and a mixed perturbation. The mixed condition applies the affine shift to a random half of adaptation examples. The main perceptron study completes 422 of 432 planned cells across three seeds. Cells missing a pretraining class are skipped. Dataset summaries average shifted conditions and seeds. Intervals bootstrap datasets. Wilcoxon tests use Holm correction within the stated families. These analyses are exploratory because earlier results on the datasets had been examined.

Keep the comparison unit fixed. Each dataset contributes one aggregated difference to the OpenML summary. This gives datasets equal weight despite different cell counts. The residual always compares a selector with the default selected by its own construction. The absolute comparison uses the untouched predictor. Compare complete procedures across constructions to separate map quality from selector learning.

Three frozen networks show small cross-fit increments. The $\mathcal{B}$ default gains 6.05 pp over the frozen perceptron. Enlarging the library improves the default by a further 0.24 pp under the cached simplicity tie rule. The cross-fitted selector then adds -0.06 pp. A deeper MLP gives 0.00 pp

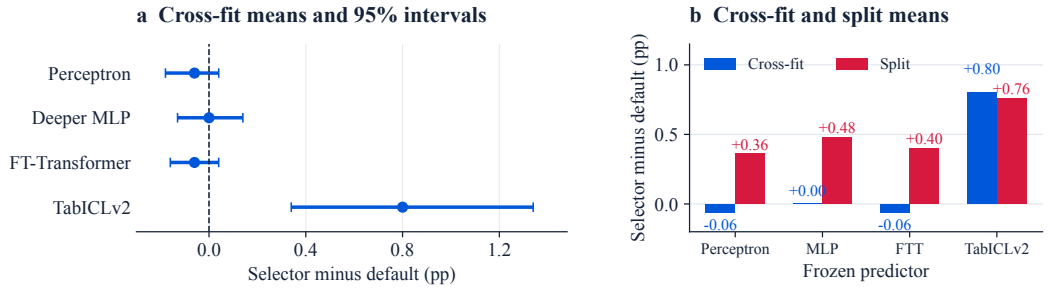

Figure 4: **Residual gains depend on both the model and the learning construction.** Left, cross-fit means and marginal 95% bootstrap intervals. Right, cross-fit and split point means. All results use 36 datasets. Network defaults use $\mathcal{B}'$. TabICLv2 uses a label-free library. The perceptron routing comparison uses simplicity tie-breaking. Split bars show reported means without fabricated uncertainty.

and FT-Transformer gives -0.06 pp. Figure 4 shows that the corresponding intervals span zero. A Brier-tie recomputation for the perceptron gives -0.03 pp with interval $[-0.11, 0.06]$. The small mean increment is stable across these two reported default rules.

A stronger residual need not mean a stronger complete procedure. Split learning gives increments of 0.36, 0.48, and 0.40 pp for the three networks. The split uses fewer examples to fit its maps. It can therefore weaken the reference that the selector must beat. In library $\mathcal{B}$, the split default loses 1.13 pp relative to its cross-fitted counterpart. The split selector loses 0.73 pp in absolute accuracy. Its residual gain alone would suggest an improvement that the complete procedure does not deliver. Both the absolute gain and the residual increment are needed to rank learning constructions.

TabICLv2 retains value for selection. TabICLv2 uses source examples as fixed context. Its candidates transform queries or context features without target labels. The default gains 1.51 pp. The cross-fitted selector adds 0.80 pp with interval $[0.34, 1.34]$. Split learning gives a similar 0.76 pp increment. This result shows that routing enlargement need not eliminate useful selection. The backbone and candidate family change together, so the comparison does not isolate either factor. Adding labeled target examples to context yields another 7.52 pp over the default. That option is effective but changes the adaptation mechanism.

5.4 Natural-shift sensitivity separates gain from inactivity

Five folktables tasks (Ding et al., 2021) use California as the source. Targets are Texas, New York, Florida, Michigan, and Puerto Rico. Each of three seeds uses 2,000 source examples. A target unit supplies 650 development, 325 guard, and 2,000 test examples. The source identifies four planned comparisons, including equivalence within ± 0.5 pp. Their registration status remains an author-reported designation, as documented in Appendix D.

Across 25 units, the default gains 2.47 pp with interval $[0.88, 4.68]$. The selector adds 0.03 pp with interval $[-0.03, 0.09]$. The reported equivalence test rejects after Holm adjustment. However, ten units contain a state-code feature, ST, that is constant in the source and protected from correction. Its target values fall far outside the source range. Every affected unit chooses identity and has zero selector increment. The aggregate therefore includes ten inactive adaptation pathways.

Figure 5 makes the sensitivity checks explicit. The 15 units without ST have a default gain of 4.12 pp and a selector increment of 0.05 pp. Restricting further to ten units without source-constant features gives 4.02 and -0.01 pp. Dropping ST and rerunning the affected ten units gives 4.73 and 0.00 pp. The rerun's selector interval is $[-0.29, 0.33]$. Removing the inactive pathway restores a useful default without producing a large residual selector gain in that rerun.

These comparisons support two distinct observations. Input adaptation can improve under natural shift. A learned selector can still add little after an effective default is available. The sensitivity analyses preserve this descriptive pattern while exposing why the original aggregate was unusually

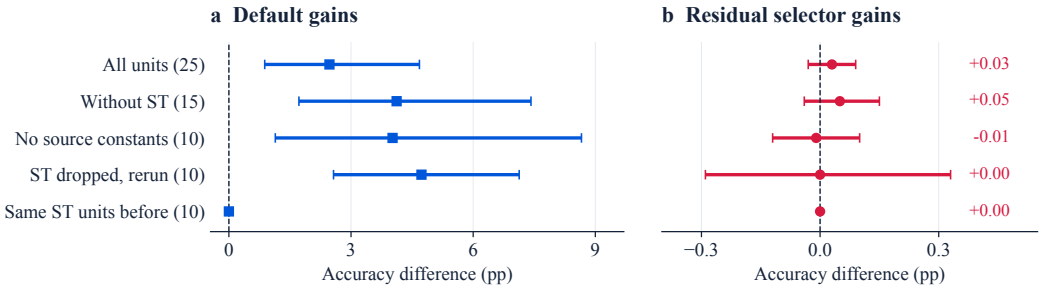

Figure 5: **The default improves accuracy while the residual selector gain stays small.** Blue measures default minus frozen model. Red measures selector minus default. Points and 95% bootstrap intervals come from Table S24. Subsets are post hoc. Dropping ST and refitting is an exploratory rerun. The last row shows the same ten ST-affected units before that correction. Rows are sensitivity views of related data, not independent replications.

Table 1: **Accuracy of the routed default minus each baseline.** Entries are mean [95% bootstrap interval] in pp. Positive values favor the $\mathcal{B}'$ default. Input baselines are reimplemented or adapted. Last-layer fine-tuning is post hoc. Values are from original Table 3 and supplement P.3.

Baseline	OpenML (36)	ACS (25)
TFM-Retouche	$+1.04$ $[0.64, 1.44]$	$+1.36$ $[0.40, 2.86]$
BETA	$+6.22$ $[4.71, 7.93]$	$+2.45$ $[0.87, 4.66]$
Last-layer retraining	$+8.35$ $[5.45, 11.48]$	$+1.60$ $[-2.16, 5.50]$
Last-layer fine-tuning	$+2.05$ $[1.44, 2.77]$	-2.82 $[-5.25, -0.72]$
Output adjustment	$+3.64$ $[2.49, 4.89]$	-0.51 $[-1.53, 0.28]$
Logistic regression	$+0.93$ $[0.09, 2.08]$	-5.60 $[-8.98, -2.52]$

concentrated near zero. Shared tasks, target units, and source data also matter for inference. The subgroup views should therefore be read together with their analysis status.

5.5 Baseline quality and calibration complete the comparison

Table 1 places the default in a wider comparison. It exceeds the reported TFM-Retouche reimplementation (Nguyen et al., 2026) and the tabular BETA adaptation (Zhang et al., 2026). Implementation behavior helps explain these results. BETA's diversity filter admits only 0.2% of examples, leaving its steering update almost inactive. Planned last-layer retraining fails in cells with extreme hidden activations. These outcomes describe the evaluated implementations. On ACS, post-hoc last-layer fine-tuning and logistic regression exceed the default by 2.82 and 5.60 pp. The frozen-input constraint therefore has an empirical cost in some settings.

Calibration adds another axis. Input corrections deliver raw frozen-model probabilities. On OpenML, the default reduces expected calibration error by 3.42 points. It reduces the Brier score multiplied by 100 by 9.13. On ACS, the calibration-error change is $+1.06$ points with interval $[-0.24, 2.21]$. Output temperature scaling reduces OpenML calibration error by another 3.85 points relative to the input default. Preserving the output interface is thus a mechanism constraint. It does not determine which method has the best probabilities.

6 Related work

Input adaptation and model reprogramming. Data fine-tuning learns image changes that improve commercial predictors (Chhabra et al., 2019). Model reprogramming also modifies inputs to reuse a frozen model (Elsayed et al., 2019; Tsai et al., 2020; Chen, 2024). Some formulations additionally remap outputs across tasks. Visual prompting learns input perturbations for pretrained models (Bahng et al., 2022). We examine the extra value of selecting among them and the routing capacity that a fair baseline should retain.

Tabular foundation models. TabICL and TabICLv2 use labeled examples as context (Qu et al., 2025; 2026). A candidate can alter query features, context features, or both. TFM-Retouche studies a lightweight input adapter (Nguyen et al., 2026). Weight fine-tuning offers a separate route. Rubachev et al. (2025) find full fine-tuning practical in their TabPFNv2 setting. A residual selector increment over label-free candidates does not establish an advantage over weight or context-label adaptation.

Abdullah (2026) study context sampling on 15 small datasets. Lee (2026) adjust test-time posteriors under label shift. TabPrep combines feature engineering with model training and tuning (Tschalzev et al., 2026). Adaptive transformation pipelines have also been studied for time-series foundation models (Qiu et al., 2026). The present comparison holds fitted transformations fixed so that library design and selector learning can be distinguished.

Distribution shift and test-time adaptation. Classical methods address covariate shift with weighted likelihood (Shimodaira, 2000). Black-box label-shift correction instead uses class-level changes (Lipton et al., 2018). Test-time adaptation covers several forms of access to unlabeled target data (Liang et al., 2025), including black-box prediction interfaces (Zhang et al., 2026). Condition-level results on CIFAR-10-C show that adaptation can help, harm, or remain inactive (Guha Majumdar & Saha, 2026). Sequential risk monitoring addresses related deployment concerns (Schirmer et al., 2025). Our guard analysis addresses a narrower setting with labeled evaluation pairs and fixed procedures.

Folktables and TableShift supply shifts tied to real data collection processes (Ding et al., 2021; Gardner et al., 2023). Conditional-label shift appears in selected tabular datasets (Liu et al., 2023). Last-layer retraining can improve robustness to spurious correlations under appropriate conditions (Kirichenko et al., 2023). FT-Transformer supplies another tabular backbone (Gorishniy et al., 2021). These comparisons motivate alternatives outside the frozen-input constraint (Table 1).

Selection and policy learning. Input-dependent selection is established in mixtures of experts (Jacobs et al., 1991), dynamic classifier selection (Cruz et al., 2018), and automated algorithm selection (Kerschke et al., 2019). Learned test-time augmentation also chooses among input transformations (Kim et al., 2020). Maximizing value over restricted policies connects to empirical welfare maximization and offline policy learning (Kitagawa & Tetenov, 2018; Athey & Wager, 2021; Zhou et al., 2023). Here, each fitted candidate's correctness can be evaluated on the same labeled example. This differs from observing only the outcome of a chosen treatment. We use value accounting to describe selection without importing guarantees that rely on the latter causal setting.

Testing the claimed improvement. Learn then Test connects risk control with hypothesis testing (Angelopoulos et al., 2025). Paired tests use disagreements to compare two procedures (McNemar, 1947). Holm correction and fixed-sequence testing address collections of claims (Holm, 1979; Westfall & Krishen, 2001). The relevant design choice here is the reference. Testing against identity and testing against a selected default answer different questions. Routing-capable baselines make that distinction consequential because they can absorb much of the total adaptation gain.

7 Conclusion

A selector's value depends on the routing already available to its baseline. Reusing fitted maps inside complete routing candidates strengthens that baseline. For fixed row-wise maps, this step preserves the pointwise oracle and cannot increase selector headroom. The controlled results show that it can absorb a large apparent selector gain. Real-data results then locate the conditions under which a residual gain remains. The practical implication is to report two quantities together. The complete procedure should improve over the untouched predictor. The selector should also justify its increment over a routing-capable default. A deployment test must use the reference required by the intended claim. These results concern the evaluated shallow selectors and libraries. Appendix C documents the deployment rules and their implementation discrepancies.

REPRODUCIBILITY STATEMENT

This reconstructed working draft is based on the supplied 12-page manuscript and 32-page supplementary PDF. It includes a self-contained definition of the main comparisons, proofs, implementation details recoverable from those PDFs, and a source-to-result map. Numerical results were transcribed and checked for documentary consistency, not rerun. The original manuscript states that code, correctness caches, manifests, and analysis plans accompany the work; those artifacts were not present in the materials reviewed for this revision. The unresolved nested-rule check is documented in Appendix C.

ETHICS STATEMENT

The reported experiments use public benchmark and de-identified survey data. The source reports no new data collection or attempted re-identification. Adapting inputs changes predictions about individuals even when model weights are unchanged. Aggregate accuracy comparisons provide no guarantee about individual decisions, protected subgroups, calibration, or deployment suitability.

AI USE STATEMENT

The supplied manuscript discloses substantial use of Anthropic Claude models through Claude Code for code, experimental design, analysis, proofs, literature review, and drafting. That historical disclosure is retained as an author-reported account, without independent verification of the underlying workflow. OpenAI Codex assisted with this revision's scientific and editorial review, primary-source bibliography checks, restructuring, LaTeX reconstruction, and figure authoring. An AI-generated composition reference informed the framework layout; the delivered scientific diagram was reconstructed as editable objects and checked against the stated mechanism. No new benchmark experiments were run in preparing this revision. Author review and responsibility for the final research claims remain necessary.

REFERENCES

M. Abdullah. Understanding Context Sampling in TabPFN on Small Tabular Datasets. *arXiv:2607.26628*, 2026. URL https://arxiv.org/abs/2607.26628.

A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei. Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998. URL https://doi.org/10.1214/24-AOAS1998.

S. Athey and S. Wager. Policy Learning with Observational Data. *Econometrica*, 89(1):133–161, 2021. doi: 10.3982/ECTA15732. URL https://doi.org/10.3982/ECTA15732.

H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola. Exploring Visual Prompts for Adapting Large-Scale Models. *arXiv:2203.17274*, 2022. URL https://arxiv.org/abs/2203.17274.

P.-Y. Chen. Model Reprogramming: Resource-Efficient Cross-Domain Machine Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 22584–22591, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/30267.

S. Chhabra, P. Majumdar, M. Vatsa, and R. Singh. Data Fine-Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8223–8230, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/view/4833.

R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti. Dynamic Classifier Selection: Recent Advances and Perspectives. *Information Fusion*, 41:195–216, 2018. doi: 10.1016/j.inffus.2017.09.010. URL https://doi.org/10.1016/j.inffus.2017.09.010.

F. Ding, M. Hardt, J. P. Miller, and L. Schmidt. Retiring Adult: New Datasets for Fair Machine Learning. *Advances in Neural Information Processing Systems*, 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/32e54441e6382a7fbacbbbaf3c450059-Abstract.html.

G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein. Adversarial Reprogramming of Neural Networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Syx_Ss05tm.

J. Gardner, Z. Popovic, and L. Schmidt. Benchmarking Distribution Shift in Tabular Data with TableShift. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a76a757ed479a1e6a5f8134bea492f83-Abstract-Datasets_and_Benchmarks.html.

Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting Deep Learning Models for Tabular Data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d86d83f925f2149e9edb0ac3b49229c-Abstract.html.

S. Guha Majumdar and A. Saha. When Test-Time Adaptation Helps, Harms, or Becomes Inactive: A Condition-Level Study on CIFAR-10-C. *arXiv:2608.22233*, 2026. URL https://arxiv.org/abs/2608.22233.

S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79. URL https://doi.org/10.1162/neco.1991.3.1.79.

P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. doi: 10.1162/evco_a_00242. URL https://doi.org/10.1162/evco_a_00242.

I. Kim, Y. Kim, and S. Kim. Learning Loss for Test-Time Augmentation. *Advances in Neural Information Processing Systems*, 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2ba596643cbbbc20318224181fa46b28-Abstract.html.

P. Kirichenko, P. Izmailov, and A. G. Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=THOOBy1uWVH.

T. Kitagawa and A. Tetenov. Who Should Be Treated? Empirical Welfare Maximization Methods for Treatment Choice. *Econometrica*, 86(2):591–616, 2018. doi: 10.3982/ECTA13288. URL https://doi.org/10.3982/ECTA13288.

S. Lee. Mitigating Label Shift in Tabular In-Context Learning via Test-Time Posterior Adjustment. *arXiv:2605.04363*, 2026. URL https://arxiv.org/abs/2605.04363.

J. Liang, R. He, and T. Tan. A Comprehensive Survey on Test-Time Adaptation Under Distribution Shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025. URL https://arxiv.org/abs/2303.15361.

Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and Correcting for Label Shift with Black Box Predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3122–3130, 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

J. Liu, T. Wang, P. Cui, and H. Namkoong. On the Need for a Language Describing Distribution Shifts: Illustrations on Tabular Datasets. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a134eaebd55b7406ff29cd75d5f1a622-Abstract-Datasets_and_Benchmarks.html.

Q. McNemar. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996. URL https://doi.org/10.1007/BF02295996.

D. Nguyen, M. Jawhar, and N. Chesneau. TFM-Retouche: A Lightweight Input-Space Adapter for Tabular Foundation Models. *arXiv:2605.06047*, 2026. URL https://arxiv.org/abs/2605.06047.

Y. Qiu, Z. Cen, Z. Pei, C. Wang, and J. Wang. Adapt Data to Model: Adaptive Transformation Optimization for Domain-shared Time Series Foundation Models. *arXiv:2603.00629*, 2026. URL https://arxiv.org/abs/2603.00629.

J. Qu, D. Holzmüller, G. Varoquaux, and M. Le Morvan. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 50817–50847, 2025. URL https://proceedings.mlr.press/v267/qu25d.html.

J. Qu, D. Holzmüller, G. Varoquaux, and M. Le Morvan. TabICLv2: A Better, Faster, Scalable, and Open Tabular Foundation Model. *arXiv:2602.11139*, 2026. URL https://arxiv.org/abs/2602.11139.

I. Rubachev, A. Kotelnikov, N. Kartashev, and A. Babenko. On Finetuning Tabular Foundation Models. *arXiv:2506.08982*, 2025. URL https://arxiv.org/abs/2506.08982.

M. Schirmer, M. Jazbec, C. A. Naesseth, and E. Nalisnick. Monitoring Risks in Test-Time Adaptation. *arXiv:2507.08721*, 2025. URL https://arxiv.org/abs/2507.08721.

H. Shimodaira. Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000. doi: 10.1016/S0378-3758(00)00115-4. URL https://doi.org/10.1016/S0378-3758(00)00115-4.

Y.-Y. Tsai, P.-Y. Chen, and T.-Y. Ho. Transfer Learning without Knowing: Reprogramming Black-box Machine Learning Models with Scarce Data and Limited Resources. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9614–9624, 2020. URL https://proceedings.mlr.press/v119/tsai20a.html.

A. Tschalzev, N. Erickson, Y. Wang, H. Rangwala, S. Lüdtke, H. Stuckenschmidt, and C. Bartelt. TabPrep: Closing the Feature Engineering Gap in Tabular Benchmarks. *arXiv:2606.02384*, 2026. URL https://arxiv.org/abs/2606.02384.

P. H. Westfall and A. Krishen. Optimally Weighted, Fixed Sequence and Gatekeeper Multiple Testing Procedures. *Journal of Statistical Planning and Inference*, 99(1):25–40, 2001. doi: 10.1016/S0378-3758(01)00077-5. URL https://doi.org/10.1016/S0378-3758(01)00077-5.

Y. Zhang, S. Niu, C. Cai, F. Liu, and J. Hamm. Adapting in the Dark: Efficient and Stable Test-Time Adaptation for Black-Box Models. *arXiv:2604.15609*, 2026. URL https://arxiv.org/abs/2604.15609.

Z. Zhou, S. Athey, and S. Wager. Offline Multi-Action Policy Learning: Generalization and Optimization. *Operations Research*, 71(1):148–183, 2023. doi: 10.1287/opre.2022.2271. URL https://doi.org/10.1287/opre.2022.2271.

A Proofs and scope of the comparisons

This appendix reorganizes the supplied supplement's theoretical arguments and implementation details. Original table identifiers refer to that supplied supplement, which remains the record for the full numerical inventory. No unseen code or result files are assumed to have been inspected.

A.1 Plug-in selection and a fixed default

Suppose $\sup_{x,k} |\hat{b}_k(x) - b_k(x)| \le \epsilon$ and $\hat{\pi}(x) \in \arg\max_k \hat{b}_k(x)$. For the conditional optimum $\pi^\star$, add and subtract estimated values:

$$b_{\pi^\star(x)}(x) - b_{\hat{\pi}(x)}(x) = [b_{\pi^\star} - \hat{b}_{\pi^\star}] + [\hat{b}_{\pi^\star} - \hat{b}_{\hat{\pi}}] + [\hat{b}_{\hat{\pi}} - b_{\hat{\pi}}] \le 2\epsilon. \tag{9}$$

Taking expectations yields $V(\pi^\star) - V(\hat{\pi}) \le 2\epsilon$. Consequently $H > 2\epsilon$ is sufficient to beat every single candidate. The implemented estimators have no established uniform bound ϵ, so this condition is not verified by the experiments.

Now fix a default g and define $a_k = b_k - b_g$. If $\sup_{x,k} |\hat{a}_k - a_k| \le \epsilon$, switching only when $\max_k \hat{a}_k(x) > \tau$ with $\tau \ge \epsilon$ guarantees a nonnegative pointwise increment over g. At a switch, the increment is greater than $\tau - \epsilon \ge 0$ and oracle regret is at most 2ϵ. Without a switch, the best increment is at most $\tau + \epsilon$. Hence regret is bounded by $\max(2\epsilon, \tau + \epsilon)$. An adaptive tree's 1.645-standard-error penalty is not a demonstrated realization of this uniform premise.

A.2 Why the reference index must agree

Suppose a reference r, the final default g, and another candidate k have sample accuracies 0.50, 0.80, and 0.70. Candidate k has a $+0.20$ increment over r but a -0.10 increment over g. Giving g the score zero while scoring k relative to r can therefore select a worse procedure. Equation (1) prevents this inconsistency. It does not imply unbiased conditional increments when the default, candidate fits, and partition were selected using the same development sample.

A.3 Paired-test validity

Under the null $p_+ \le p_-$, conditioning on the number of discordant pairs gives

$$n_+ \mid m \sim \mathrm{Bin}(m, q), \qquad q = \frac{p_+}{p_+ + p_-} \le \frac{1}{2}.$$

The rejection set of Equation (8) is an upper-tail set. Its probability is largest at $q = 1/2$, where discreteness makes it at most α. Averaging over m preserves the bound. No discordances imply no rejection. This argument requires independently sampled row-level pairs and procedures fixed before the guard sample is evaluated.

A development-selected procedure is fixed conditional on development data, so one valid test against identity controls false adoption relative to that comparator. Holm correction controls family-wise error for a pre-specified collection of valid null tests. A strict two-stage fixed sequence controls family-wise error because rejection of any true null requires rejection of the first true null in the order, tested at level α. This statement does not settle what an implementation does when its first reference and candidate are exactly the same procedure.

B Implementation specification recovered from the supplement

B.1 Data and frozen networks

OpenML preprocessing keeps numeric columns, removes columns with more than 50% missing values and classes with fewer than 20 examples, and caps datasets at 2,000 examples by stratified subsampling with seed 12345. Eligibility requires at least 500 examples and between three and 100 features. Pretraining uses 35% of examples; the remainder is divided 50:25:25 into development, guard, and test sets. Imputation and standardization use source statistics.

The baseline called a "perceptron" is an MLP with two hidden layers of 128 units, not a single linear perceptron. The deeper MLP has four layers of 256 ReLU units. The supplement specifies 250 full-batch epochs, AdamW with learning rate 10^{-3} and weight decay 10^{-4} for the deeper model. FT-Transformer is reimplemented with three pre-norm blocks, dimension 192, eight heads, ReGLU feed-forward layers, attention dropout 0.2, and feed-forward dropout 0.1. It uses AdamW at 10^{-4}, weight decay 10^{-5} with the specified exclusions, batch size 256, and at most 200 epochs with

patience 16 on a source-validation fifth. Standardization replaces quantile transformation so the input map remains differentiable.

Selection bias samples source examples from the lower half of the first principal component of imputed, standardized features. The affine condition perturbs continuous adaptation features, including a partial rotation of strength 0.35. The mixed condition applies the same perturbation to a random half of adaptation examples. Neither the PDFs nor this reconstruction supplies the dataset manifest or cached arrays needed to verify the exact realized samples.

B.2 Libraries and selectors

The real-data network library sizes are 6, 10, and 16 for $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{B}'$. The synthetic $\mathcal{A}$ contains identity, translation, affine, and low-rank maps; synthetic $\mathcal{B}$ adds cluster-fitted low-rank maps. The enlargement result always concerns maps shared *within* a comparison, not numerical equality between separate diagnostic runs.

Learned maps use Adam with learning rate 10^{-2} and 80 steps. Their initial identity state is retained unless an inner validation quarter strictly improves. Two-cluster k-means is computed from all development features before map fitting. The cross-fit construction uses five folds and a final refit. The split construction halves development data with label stratification, cross-fits candidates on D_1, refits on D_1, and learns the selector from D_2. Because clustering already used D_2 features and the split depends on labels, an independence claim does not follow from this separation of fitting labels.

The depth-two tree considers nine thresholds per feature at node quantiles from 10% to 90%. It greedily maximizes the sum of each leaf's largest unpenalized paired gain, including zero for the default. A leaf uses its best candidate only if the corresponding mean increment minus 1.645 standard errors is positive. The alternative per-candidate selector fits one depth-three regression tree per candidate with minimum leaf size $\max(20, n/20)$; it is a secondary comparison.

The supplement also mentions temperature fitting on a grid from 0.05 to 20 inside fitting subsets, while explicitly stating that downstream input-correction probabilities are the raw frozen outputs. A faithful implementation must separate such an auxiliary fitting loss from delivered probabilities. The PDFs alone do not expose that code path, so the reconstructed specification preserves the downstream restriction without claiming a code-level verification.

B.3 TabICLv2

The supplied specification identifies package `tabicl 2.0.2`, checkpoint `tabicl-classifier-v2-20260212`, CPU inference, eight ensemble members, and a fixed random state. The labeled source rows provide context. Label-free candidates match query means or moments to context statistics, transform context moments toward development statistics, or standardize both context and queries. Context labels stay fixed. Candidate statistics can still depend on development features even though no labels fit these maps.

The batch probe compares predictions on the same test rows in a full batch, chunks, and single-row calls. One of 36 datasets changes predicted labels, and the reported accuracy difference reaches 4.9 pp. The probe does not cover every candidate. Thus the row-wise guard premise is not established, and context/query-batch transformations should not be silently covered by Proposition 1 either.

B.4 Controlled generator and diagnostic evaluation

Let $Z \sim \mathcal{N}(0, I_4)$, let w be proportional to $(1, 0.6, 0.3, 0)$, and sample $Y \sim \mathrm{Bernoulli}(\sigma(3w^\top Z))$. Group-specific rotations act in the plane containing w and an orthogonal direction. The site feature has distribution $U \mid g \sim \mathcal{N}(\mu_g, 1)$ with means $\pm\delta$. Rotations preserve the isotropic Gaussian feature distribution; group information comes from U.

For a target feature vector x,

$$P(Y = 1 \mid x) = \sum_g P(g \mid u)\, \sigma(3w^\top R_g^\top x_z).$$

This allows evaluation of conditional accuracy without sampling additional labels. The main tables average 10^5 feature draws per repetition. The value-accounting diagnostic uses two independent banks

of 50,000 features, evaluates fixed trees and maps, and compares pooled plug-in oracle actions with actions chosen on one bank and evaluated on the other. The reported precision trigger for increasing to 200,000 draws per bank fired in no setting. Bootstrap intervals across ten fitted repetitions reflect training variation, separately from Monte Carlo error.

C Deployment-rule audit requiring implementation evidence

The source text defines nested adoption as testing $\hat{g}$ against identity and, only after rejection, testing the selector against $\hat{g}$. Its Proposition 3 explicitly states that $m = 0$ does not reject. Hence a strictly implemented first stage cannot pass when $\hat{g} = \mathrm{id}$.

The fixed-library identifiable setting has best-single gain zero in original Table S19. Yet original Table S3 reports a mean nested deployed gain of 12.1 pp, and Table S14 reports 7.0, 11.9, 13.8, and 14.3 pp at total label budgets 300, 600, 1200, and 2400, while the single-only procedure remains at zero. These entries do not follow naturally from the literal strict-gate description. For example, if all fixed candidates have nonpositive population benefit and pointwise headroom is approximately 14.9 pp as reported, a valid strict first stage passes with probability at most 0.05. Its expected deployed gain is then at most approximately $0.05 \times 14.9 = 0.745$ pp. A finite sample average is not itself an expectation, but the scale of the discrepancy demands reconciliation.

One possible explanation is an explicit identity shortcut: if the chosen default is identity, directly test the selector against identity. Such a rule can be sensible, but it no longer establishes that the default strictly improves over identity. Other possibilities include a different gate, a result-label mismatch, or dependence invalidating the stated test. The PDFs cannot distinguish these explanations, and no particular explanation is asserted here.

The audit needs, for each repetition, the selected default, its population value, the first-stage discordant counts and p-value, any shortcut condition, the second-stage comparator and p-value, and the deployed action. Until those records are available, the original learned-library comparison of 70% versus 15% withholding should be treated as a reported result of an unresolved implementation rather than definitive evidence about the written strict rule.

D Evidence map and appendix organization

Table 2 keeps the final narrative traceable to the complete source supplement. The source's earlier round used inconsistent references for selector increments and is a provenance record, not a current baseline. Correcting that issue changes the learning target but does not establish that all downstream code is correct.

The two supplied PDFs contain no executable experiment package, manifest, per-cell caches, source hashes to compare with files, or original analysis-plan files. References to such artifacts establish what is claimed to exist, not that it was delivered or independently checked. The package accompanying this revision contains the rewritten text, checked bibliography, plotting data transcribed from the source tables, and editable figures. It preserves the original source PDFs separately so that no historical table is silently discarded.

For a submission version, the supplement should be ordered by reader need: proofs; data and fitting details; main extended results; diagnostics; natural-shift sensitivity; baseline implementations; and provenance/reproduction records. Experiment-round names and implementation history belong in the final provenance section. A timestamped analysis plan written before computation is a pre-specified analysis; the stronger designation "pre-registered" requires the actual registration/provenance evidence to be inspectable. The source's natural-shift status is retained as an author-reported designation rather than independently certified here.

Table 2: Source-to-revision map. Original table identifiers refer to the supplied 32-page supplement.

Evidence	Original location	Treatment in this revision
Plug-in, pairing, guard and routing results	A, pp. 1–3	Proofs and explicit scope
Fitting, preprocessing, model details	B–D, pp. 3–5	Reconstructed implementation appendix
Version history and prior dataset exposure	E, p. 6	Exploratory status retained
Synthetic library and routing effects	S1–S2, S8	Main controlled study and Figure 2
Adoption and fixed label budgets	S3, S14, S23	Nested-rule audit before interpretation
Network comparisons and adoption rates	S4–S7, S17	Residual and absolute values separated
Refitting diagnostic	S9–S10	Limited negative diagnostic; no causal claim
Value accounting	S11	Loss identity with oracle caveats
Split-score and search diagnostic	S12–S13	Negative repair result retained
TabICLv2 and batch dependence	S15–S16	Positive increment with no exact guard claim
Objective and historical results	S18–S23	Source archive; not promoted as new method
Natural shift and state-code sensitivity	S24–S25	Defect and analysis status foregrounded
Additional models and tie rules	S26	Configuration distinctions retained
Baselines and calibration	S27–S28	Implementation and scope caveats retained
Reproduction commands	Q, p. 30	Inventory only; not evidence of execution