What Source Calibration Buys Under Real Tabular Shift

Anonymous authors
Paper under double-blind review

Abstract

Real tabular shift breaks the conformal coverage guarantee of tabular foundation models (TFMs). Once a rule reuses the source calibration set, its tail coverage at the same nominal level need not reflect better calibration, since a higher threshold buys coverage with width. We study what the source calibration set is worth for several TFMs on real shifts from US Census, wine and UCI tables. We propose a width-matched control, a reporting protocol that calibrates on the target labels alone at an inflated quantile rank, matches each rule on its mean set size, and summarizes coverage per calibration draw with paired cluster bootstraps over settings and tasks. For source-anchored rules, a bias-variance comparison gives a crossover budget below which the source quantile has lower risk, and it orders the observed crossover across settings, though retrospectively. This supports equal-width comparisons of pooling, prior and gated rules. Experiments show that nominal-level reporting is optimistic for the gated rules, whose gains are mostly width: against the control, a per-draw gain survives at small label budgets and fades as labels grow, pooling rules fall below the control, and the closest prior method sits close to it. Label-free reweighting does not recalibrate real shift under stronger density-ratio estimators or label-shift weights, and the same holds for gradient boosting and logistic regression. Code is available at `https://anonymous.4open.science/r/TFM-1-ICLR2027-576E/`.

1 Introduction

A tabular foundation model (TFM) takes a labeled support set and a query row and predicts the query's label in one forward pass, with no per-dataset training (Hollmann et al., 2023; Qu et al., 2025; 2026), at the accuracy of tuned gradient boosting (Grinsztajn et al., 2022; McElfresh et al., 2023). Conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023) turns such a predictor into prediction sets with a coverage guarantee, provided the calibration set and the test point are exchangeable. In deployment they rarely are. A model calibrated on one state, year or population is used on another, which is the rule in credit, healthcare and public policy (Gardner et al., 2023; Ding et al., 2021), and the guarantee is then void: a model that is 92% confident and right 70% of the time on a new state is worse than one that abstains. Two remedies compete. Covariate-shift conformal (Tibshirani et al., 2019) reweights the source calibration set by the density ratio between target and source covariates; it needs no target labels and is exact when only $P(X)$ moves. Few-shot target calibration recomputes the quantile from a small labeled target sample; it is valid under any shift, but its quantile has variance of order $1/k$ and is unreliable until k is in the hundreds.

Beyond either remedy alone, a growing set of rules reuses the source calibration set together with the labeled target points: pooling the two samples, pooling them at density-ratio weights (Wang & Qiao, 2025), down-weighting the source points (Barber et al., 2023), or anchoring the target quantile to the source quantile. Once a rule reuses the source in this way, the number reported for it is ambiguous. A rule that returns a higher threshold covers more of the target and also returns wider sets, so a gain in tail coverage over target-only calibration at the same nominal level need not mean that the source improved the calibration. It may mean only that the rule is more conservative. Evaluation must therefore distinguish two questions. A nominal-level question asks whether the rule covers more than target-only calibration at the level both were asked for. A width-matched question asks whether it covers more than target-only calibration that returns sets of the same size. The two coincide only when the rule returns sets as wide as target-only calibration does. We refer to this gap as width

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.

mistaken for reliability, and it leads us to treat coverage at matched set size, per calibration draw, as the object of evaluation.

Recent work makes the source-reuse setting concrete but does not measure it this way. Weighted conformal and its descendants (Park et al., 2022; Qiu et al., 2023; Yang et al., 2024; Cauchois et al., 2024) assume that $P(Y \mid X)$ is shared and are validated on synthetic or semi-synthetic covariate shift. Methods that pool source and labeled target points (Wang & Qiao, 2025) or use unlabeled target data (Correia & Louizos, 2025; Siahkali et al., 2026; Min et al., 2026) report coverage against target-only calibration at the same nominal level. We ask a different question. Specifically, after a rule has reused the source calibration set, what does the source buy over target-only calibration at equal width, and does any label-free reweighting rule recalibrate real tabular shift at all? Appendix B extends the related work.

Motivated by these questions, we propose a width-matched control for evaluating rules that reuse source calibration data. The control is target-only calibration at an inflated quantile rank, a one-parameter family that trades width for coverage on the target labels alone. For each rule we read the control at the rule's own mean set size, summarize coverage by its tenth percentile over independent calibration draws, so that draw-to-draw variance is scored rather than averaged away, and attach paired cluster-bootstrap intervals that resample settings or whole tasks with the matching recomputed inside every replicate. To measure what the source is worth at best we also use a probe, source-anchored shrinkage conformal (SASC), which takes the source quantile, corrects its location on the target labels, and admits the anchor only when a held-out target check passes. SASC is the best case for the source that we could construct, not a proposal. This is the reporting layer analyzed in Section 3.

We validate the control and the probe theoretically and empirically. For source-anchored rules, a bias-variance comparison of the source and few-shot quantiles gives a crossover budget $k^\star = \kappa/b^2$ below which the anchor has lower risk; it orders the observed crossover across settings but needs target densities, so it is retrospective. Experiments on five frozen TFMs from three pretraining lineages, 46 US Census and wine shifts and 11 UCI shifts show that the two gated rules beat target-only calibration at the nominal level, that the control removes most of that gain, that pooling rules and the closest prior method fall below target-only calibration with or without the control, and that no label-free reweighting rule recalibrates real shift under four density-ratio estimators or label-shift weights. The same holds for XGBoost and logistic regression.

Key contributions.

- We formulate the evaluation of rules that reuse source calibration data as a width-matched comparison, and identify coverage at matched set size, per calibration draw, rather than tail coverage at the nominal level, as the target of evaluation.
- We propose a width-matched control based on target-only calibration at an inflated rank, size matching, per-draw tail statistics and paired cluster bootstraps over settings and tasks, together with a probe, SASC, that uses the source as well as we can.
- We derive a crossover budget $k^\star = \kappa/b^2$ for when a source-anchored quantile has lower risk than the few-shot quantile, and show that it orders the observed crossover across 46 settings and predicts the failure of anchoring on cumulative-mass scores.
- We validate the control on five TFMs and two classical models over 57 real shifts, showing that source-reuse gains are mostly width, that label-free reweighting does not recalibrate real shift, and that the per-cell regime the criterion locates is where a small gain survives.

2 Problem formulation

We first define the target of interest. The setup covers classification and regression with any split-conformal score. Let P_S denote the source law of (X, Y) and P_T the target law. A frozen TFM f maps a support set of n_{tr} source rows and a query x to a predictive distribution $\hat{p}(\cdot \mid x)$, or to a point prediction $\hat{y}(x)$ for regression. A nonconformity score $s(x, y)$ measures how poorly the output fits the label; we use the least-ambiguous-set (LAC) score $s(x, y) = 1 - \hat{p}(y \mid x)$ for classification and the absolute residual $|y - \hat{y}(x)|$ for regression. Split conformal calibration takes the scores $s_1, \ldots, s_n$ of a calibration set and returns the set

$$C_q(x) = \{y : \ s(x, y) \le q\}, \qquad q = Q_{1-\alpha}(s_1, \ldots, s_n), \tag{1}$$

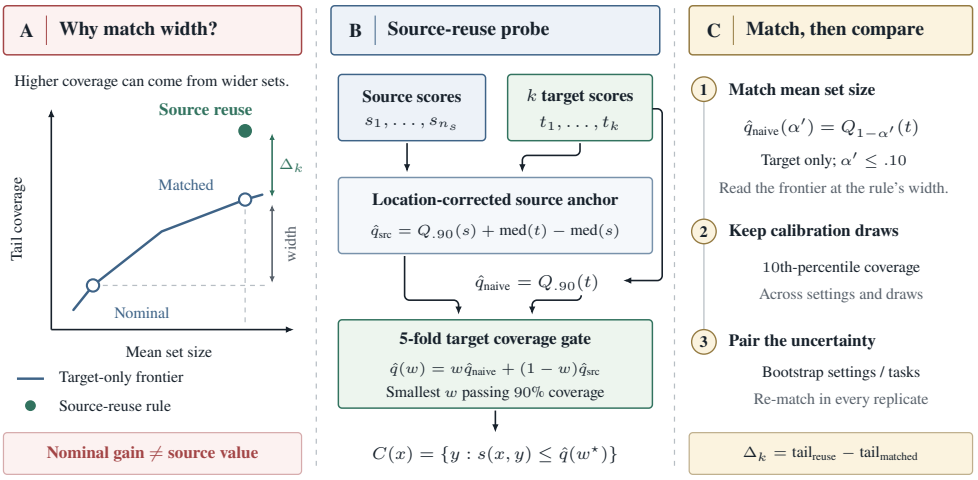


Figure 1: Overview. (A) Why width is matched: a rule that returns wider sets gains tail coverage without better calibration, and Δ_k is what remains at the rule's own set size. (B) The probe: a location-corrected source anchor blended with the target-only quantile under a held-out coverage gate. (C) The control: match on mean set size, keep calibration draws, and pair the uncertainty by re-matching inside every bootstrap replicate.

where $Q_{1-\alpha}$ is the $\lceil (n+1)(1-\alpha) \rceil$-th order statistic. When the calibration scores and the test score are exchangeable, $P(Y \in C_q(X)) \ge 1-\alpha$. We set $\alpha = 0.1$ throughout. Under shift the n_s source calibration scores $\{s_i\}$ are not exchangeable with target scores and (1) carries no guarantee.

A calibration rule under shift is a map from the source scores $\{s_i\}$, a labeled target sample $\{t_j\}_{j=1}^k$ of k scores, and possibly unlabeled target covariates, to a threshold $\hat{q}$. Two rules use one input only. Weighted conformal (Tibshirani et al., 2019) uses the source scores at weights $w(x) = \mathrm{d}P_T/\mathrm{d}P_S(x)$, estimated by a domain classifier, and needs no target labels. Naive few-shot calibration uses the target scores only, $\hat{q}_{\text{naive}} = Q_{1-\alpha}(t_1, \ldots, t_k)$; it is valid under any shift but has variance of order $1/k$. The rules of interest use both inputs and are defined in Section 3.2.

For a rule with threshold $\hat{q}$ we record on a target test set its coverage $\text{cov}(\hat{q}) = P_T(Y \in C_{\hat{q}}(X))$ and its mean set size $\text{size}(\hat{q}) = \mathbb{E}_T|C_{\hat{q}}(X)|$. Over a collection of settings, each a source, a target and a base model, and calibration draws, each a draw of the k labels, the tail coverage is the 10th percentile of $\text{cov}(\hat{q})$, taken per setting after averaging draws or per draw. The value of the source for a rule at budget k is its tail coverage minus the tail coverage of target-only calibration that returns the same mean set size,

$$\Delta_k = \text{tail}\big[\text{cov}(\hat{q})\big] - \text{tail}\big[\text{cov}(\hat{q}_{\text{naive}}(\alpha'))\big] \quad \text{with} \quad \text{size}(\hat{q}_{\text{naive}}(\alpha')) = \text{size}(\hat{q}), \tag{2}$$

where $\hat{q}_{\text{naive}}(\alpha') = Q_{1-\alpha'}(t_1, \ldots, t_k)$ is target-only calibration at a level $\alpha' \le \alpha$. Thus Δ_k is the coverage the source buys over spending the same width on target labels alone, not the gain over target-only calibration at the nominal level. Section 3 defines the protocol used to estimate (2) with confidence intervals.

3 The width-matched control

The width-matched control is our protocol for reporting what a rule that reuses the source calibration set is worth. It takes the rule's deployed threshold, the target labels used to build it and the target test set, and asks one question: at the width the rule returns, how much coverage do target labels alone provide? The protocol has three parts, a size-matched comparator, a per-draw tail statistic and a paired cluster bootstrap. Figure 1 gives an overview together with the probe of Section 3.2.

3.1 Protocol

We first fix the comparator. Target-only calibration at level α' returns $\hat{q}_{\text{naive}}(\alpha') = Q_{1-\alpha'}(t_1, \ldots, t_k)$. Lowering α' raises the rank, so over the grid $\alpha' \in \{0.10, 0.09, \ldots, 0.02\}$ the comparator traces a

frontier of mean set size against tail coverage built from target labels alone. For a rule with mean set size $\bar{s}$ we interpolate the frontier at $\bar{s}$ and report the difference (2). Matching on mean coverage instead of size gives a second reading, which we report alongside (Appendix X.1).

We next fix the statistic. Naive few-shot calibration is marginally valid on average, but its realized coverage has standard deviation $\sqrt{\alpha(1-\alpha)/k}$, about four points at $k{=}50$, so the quantity a deployment experiences is the coverage of one calibration draw, not the average over draws. For each setting, seed and budget we draw the k labels ten times independently from the target pool and compute every rule on every draw. The per-draw tail is the 10th percentile of coverage over all settings and draws; the per-setting tail averages draws first and takes the 10th percentile over settings.

Finally we attach uncertainty. The 46 census and wine settings share one source and five tasks, so a bootstrap has to resample dependent blocks. We resample settings, or whole tasks, with replacement, and recompute the frontier, the match and the difference (2) inside every replicate, two thousand times for the probe and one thousand for the other rules. A gain is one whose interval excludes zero under both block structures.

3.2 The probe and the rules

To measure what the source is worth at best we need a rule that uses it as well as we can; SASC is that probe. Let $\{s_i\}$ be the n_s source calibration scores and $\{t_j\}$ the k target scores. The source anchor is $\hat{q}_{\text{src}} = Q_{1-\alpha}(\{s_i\}) + \big(\text{med}(\{t_j\}) - \text{med}(\{s_i\})\big)$, the source quantile shifted by the median gap estimated on the target. It has low variance, since k points estimate a median more stably than a $(1-\alpha)$ quantile, and it is biased by $b = (q_S - m_S) - (q_T - m_T)$, the difference of the source and target median-to-tail spreads, whenever the shift is not a translation (Appendix U). SASC shrinks the naive quantile toward the anchor,

$$\hat{q}_{\text{SASC}}(w) = w\, Q_{1-\alpha}(\{t_j\}) + (1-w)\, \hat{q}_{\text{src}}, \tag{3}$$

and chooses w on an eight-point grid as the smallest value whose five-fold held-out coverage on the target labels is at least $1-\alpha$; if none passes, $w{=}1$ and the rule is naive. The gate is one-sided. An anchor that under-covers fails the check and SASC reverts to naive, whereas an anchor that over-covers passes and is kept, so the limiting threshold over-covers when $b > 0$ (Appendix D.2). Selecting w on the target labels breaks exchangeability, and the point rule carries no finite-sample guarantee; a conservative form that floors the threshold at a training-conditional target quantile is valid at a set-size cost (Appendix D.2). Conditional anchoring runs SASC within tertile bins of the model's own confidence, each bin anchored to its own source scores, with a fallback to the marginal rule below eight labels. The cost of the probe is one sort and a five-fold scan over at most 500 scalars (Algorithm 1, Appendix C).

We compare the probe with four rules that also reuse the source:

- Pooled takes the plain quantile of the location-shifted source scores together with the target scores.
- Pooled-DR is weighted conformal over the source scores at their density-ratio weights and the target scores at weight one.
- WCC-CSPD (Wang & Qiao, 2025), the closest prior method, is a class-wise weighted conformal classifier that pools source and labeled target points at class-conditional density-ratio weights (Appendix L).
- NexCP-gated is non-exchangeable conformal (Barber et al., 2023) with one data-independent weight ρ on every source point, chosen by the same held-out coverage check as SASC's w; it separates what is transferred from how much.

For the label-free route we temper the density-ratio weights, $w \mapsto w^{\tau}$, and select τ without target labels. AUTO-τ splits the source calibration set, estimates the target coverage of each candidate τ by importance weighting on the held-out half, and takes the smallest τ whose lower confidence bound, calibrated by the normalized effective sample size $n_{\text{eff}} = (\sum w_i)^2/(n \sum w_i^2)$, reaches $1-\alpha$ (Algorithm 2, Appendix I; consistent under covariate shift, Prop. 7). We also report the temper closest to nominal when chosen with test labels, the oracle temper, and the maximum-coverage temper. Their gaps over plain conformal bound what any temper on the grid can do.

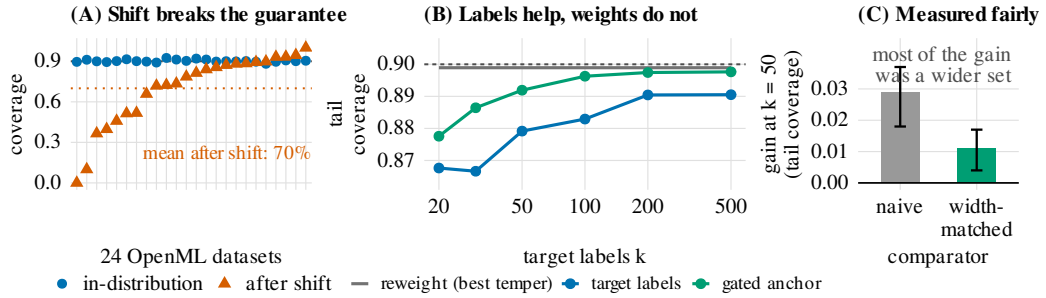


Figure 2: Summary of the findings (TabICLv2). (A) Coverage per OpenML dataset in distribution and after shift. (B) Tail coverage of the three routes against the label budget on the 46 real settings. (C) The probe's gain at k=50 against naive and against the width-matched control.

3.3 When the source can help: a crossover budget

We now state what the probe can and cannot buy, in the regime where quantile estimators are asymptotically normal. Let f_T be the target score density, m_T its median and q_T its $(1-\alpha)$ quantile.

Proposition 1 (Crossover budget, informal). *As $k \to \infty$, $\mathrm{Var}(\hat{q}_{\text{naive}}) = \alpha(1-\alpha)/(k f_T(q_T)^2) + o(1/k)$ and $\mathrm{Var}(\hat{q}_{\text{src}}) = 1/(4k f_T(m_T)^2) + O(1/n_s)$, while the anchor carries the bias b. The anchor has lower mean squared error than the naive quantile if and only if*

$$k < k^\star = \kappa/b^2, \qquad \kappa = \frac{\alpha(1-\alpha)}{f_T(q_T)^2} - \frac{1}{4 f_T(m_T)^2}. \tag{4}$$

The proof is standard quantile asymptotics and is given with the formal conditions in Appendix D.3 (Prop. 4). Proposition 1 explains where the source can pay and where it cannot. The anchor helps only below a budget set by the difference of two variances over a squared bias, and $\kappa > 0$ requires $f_T(m_T)/f_T(q_T) > 1/(2\sqrt{\alpha(1-\alpha)}) = 1.67$ at $\alpha = 0.1$: the score density must be sparse at the quantile relative to the median. LAC scores satisfy this, since most of their mass sits near zero, whereas cumulative-mass scores (APS, RAPS) have a ratio near one, so $\kappa < 0$ and the anchor cannot help at any budget. A larger bias b lowers $k^\star$, so a more severe shift shortens the regime. Two limits set the scope. The quantities in (4) are target densities and a target bias, so the budget is a diagnostic computed after the fact rather than a rule for choosing k in advance; from the source scores and k labels alone the ordering of settings is only partly recovered (Appendix U). And $k^\star$ is about threshold risk, whereas the gate targets coverage; the two disagree once the anchor over-covers, which is why the probe's coverage gain does not track $k^\star$ (Section 4.3).

4 Experiments

Rules that reuse the source calibration set look better than target-only calibration at the nominal level, and Section 2 argued that this comparison conflates calibration with width. Our experiments ask whether the source improves calibration once width is matched, whether the label-free route recalibrates real shift at all, and whether the crossover budget explains where a gain survives. We organize them around three research questions:

(RQ1) At matched width, what does reusing the source calibration set buy over target labels alone?
(RQ2) Does any label-free reweighting rule recalibrate real tabular shift?
(RQ3) Does the crossover budget explain when the source pays, and is that regime usable?

We evaluate five frozen TFMs from three pretraining lineages: TabICLv2 (Qu et al., 2026), TabPFNv2 (Hollmann et al., 2025) and TabPFNv3 (Grinsztajn et al., 2026), amortized over synthetic priors; TabDPT (Ma et al., 2025), pretrained on real tables with a retrieved context; and GAMformer (Mueller et al., 2024), which emits the shape functions of an additive model. Real shift comes from five person-level US Census tasks via folktables (Ding et al., 2021), income, employment, public coverage, mobility and travel time, with California 2018 as the source and eight other states in 2018 plus

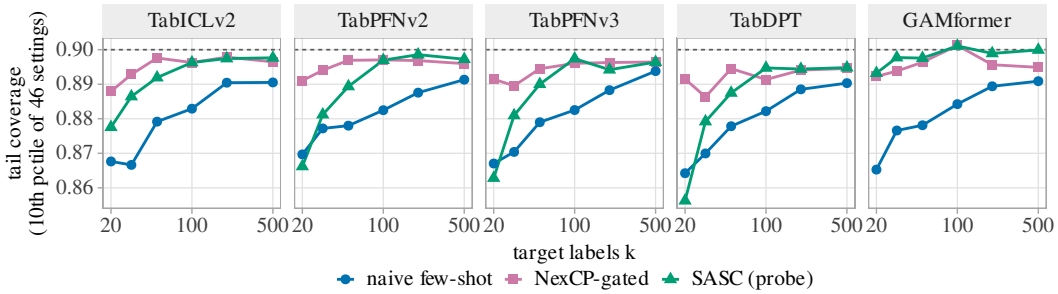


Figure 3: Tail coverage against label budget on the 46 census and wine settings, one panel per TFM. Dashed line: nominal 0.90.

California 2014 as targets, and from the wine quality data of Cortez et al. (2009), red as source and white as target; together these are the 46-setting suite. A second family of 11 settings comes from six public UCI tables with documented domain splits in the manner of TableShift (Gardner et al., 2023), hospital readmission, forest cover, electricity, bank marketing, Adult income and a drifting gas-sensor array, with the split feature removed from the predictors (Appendix S). Support and calibration sets of 3000 rows each are drawn from the source, the labeled budget $k \in \{20, 30, 50, 100, 200, 500\}$ from a 500-point target pool, and coverage is measured on a disjoint 2500-point test set. Every rule is recomputed from one cached set of base-model predictions per setting and seed, so all comparisons are exactly paired, with 5 seeds; the label-free temper and few-shot tables that re-run the base model use 3 seeds. Unless otherwise stated the control uses 10 draws per budget, the 9-point rank grid of Section 3.1 and 2000 bootstrap replicates (Appendix F).

Appendix H, I and D.5 provide the controlled studies behind this setup. Under an induced covariate split of 24 OpenML datasets, split-conformal coverage of every TFM falls from 0.90 to between 0.697 (TabICLv2) and 0.775 (TabPFNv2, GAMformer), a spread wider than the gap between lineages, and the loss is broad rather than driven by a few datasets (Figure 2A). Under controlled covariate shift on the same datasets, AUTO-τ tracks the oracle temper to within 0.015, so a good temper exists and can be found without labels when the shift is covariate. Three attempts to identify a conditional shift on the real settings directly failed their own control, so we do not claim that the residual measured below is concept shift.

4.1 RQ1: What the source buys at matched width

We first answer RQ1 by comparing every rule with target-only calibration at the nominal level and then with the width-matched control. At the nominal level both gated rules lift tail coverage above naive at every $k \geq 30$ for all five TFMs, significantly at $k \geq 50$ (Figure 3; Appendix X), whereas pooling, density-ratio pooling and WCC-CSPD fall below naive for the four unconstrained TFMs at every budget (Appendix L). The gated rules also raise mean coverage from 0.900 to 0.917 and set size by 3%, which is the pattern of a higher threshold.

Table 1 reports the difference (2) per draw. Pooling and density-ratio pooling stay below the control at k=50 for four of five TFMs, since they return narrower sets than naive and lose coverage with them. WCC-CSPD is within 0.013 of the control at k=50. The two gated rules keep a per-draw gain, +0.024 to +0.047 at k=20 and +0.011 to +0.024 at k=50, NexCP-gated's larger than SASC's and still +0.002 to +0.005 at k=200 where SASC's is negative. For SASC alone the gain is +0.018 to +0.022 at k=30, indistinguishable from zero at k=100 and negative from k=200. Percentiled over settings rather than draws, the control matches or beats SASC in every cell, by 0.003 to 0.008 at k=50 when matched on size and by up to 0.028 when matched on coverage (Appendix X.1). The paired intervals exclude zero for SASC at $k \leq 50$ for every TFM under setting blocks and for four of five at k=50 under task blocks, and for NexCP-gated at $k \leq 100$ under both; they include or fall below zero for SASC from k=100 (Appendix X.1).

Two further checks bound what the gated rules offer. A block bootstrap over ACS tasks gives 95% intervals over naive of $[+0.003, +0.028]$ for SASC, $[+0.007, +0.029]$ for NexCP-gated and $[-0.047, -0.001]$ for WCC-CSPD (TabICLv2; Appendix X): ungated rules transfer whatever the weights miss as source bias, and the held-out gate is what turns the source into a small gain (the un-

Table 1: Per-draw gain over the width-matched control at k=20 and k=50, 46 settings, 5 seeds, 10 draws. Bold marks a gain; last column, set size relative to naive.

k	rule	TabICLv2	TabPFNv2	TabPFNv3	TabDPT	GAMformer	size vs. naive
20	SASC (gated anchor)	**+0.028**	**+0.029**	**+0.029**	**+0.024**	**+0.026**	+1%
	NexCP-gated	**+0.043**	**+0.046**	**+0.047**	**+0.043**	**+0.047**	+1%
	Pooled	**+0.009**	-0.026	-0.036	-0.073	**+0.026**	-3%
	Pooled-DR	**+0.005**	-0.008	-0.012	-0.030	**+0.049**	-7%
	WCC-CSPD	**+0.018**	**+0.018**	**+0.015**	**+0.008**	**+0.037**	-2%
50	SASC (gated anchor)	**+0.011**	**+0.015**	**+0.011**	**+0.013**	**+0.013**	+3%
	NexCP-gated	**+0.021**	**+0.024**	**+0.022**	**+0.022**	**+0.022**	+2%
	Pooled	-0.006	-0.059	-0.074	-0.108	**+0.010**	-2%
	Pooled-DR	-0.018	-0.030	-0.036	-0.049	**+0.023**	-6%
	WCC-CSPD	-0.005	-0.007	-0.010	-0.013	**+0.003**	-0%

Table 2: Four density-ratio estimators on real ACS shift, 46 settings, five TFMs pooled, 5 seeds; n_{eff} is the normalized effective sample size of the weights.

	plain		full weight		AUTO-τ		best-calibrated		max coverage		
ratio estimator	cov	size	cov	size	cov	size	cov	size	cov	size	n_{eff}
logistic (used above)	0.880	1.36	0.917	1.53	0.920	1.53	0.897	1.40	0.929	1.55	0.67
GBDT, cross-fitted	0.880	1.36	0.932	1.59	0.922	1.54	0.906	1.43	0.943	1.62	0.44
GBDT, cross-fitted, isotonic	0.880	1.36	0.902	1.54	0.917	1.53	0.904	1.42	0.938	1.60	0.63
TabPFNv2, cross-fitted	0.880	1.36	0.937	1.61	0.926	1.55	0.903	1.42	0.943	1.62	0.37

gated schedule $w_k = k/(k + K_0)$ falls below naive at large k, Appendix P). The conservative form of SASC is certified at a width cost (1.66 against 1.46 at k=50), but certified naive, Clopper–Pearson on the target alone, is within 0.01 of it (Appendix Y), and with a 50k-row source the anchor is biased and the gate detects it, matching naive for $k \geq 30$ (Appendix G). These results answer RQ1: the source buys draw-to-draw stability at the smallest budgets, worth one to five points of per-draw tail coverage at $k \leq 50$ for the two gated rules, and nothing at matched width beyond about k=100; pooling and the closest prior method buy nothing at any budget. Most of the apparent gain at the nominal level is width.

Takeaway 1

Reusing the source calibration set buys width more than calibration: at matched set size only the gated rules keep a gain, one to five points of per-draw tail coverage at $k \leq 50$, while pooling and the closest prior method buy nothing at any budget.

4.2 RQ2: Label-free reweighting on real shift

We next answer RQ2 by asking whether any temper of the density-ratio weights recalibrates the five census tasks. On the five ACS tasks the temper closest to nominal barely exceeds plain conformal, 0.899 against 0.882 for TabICLv2 and 0.897 against 0.871 for TabPFNv2 (Table 17). The maximum-coverage temper reaches 0.931, but at set size 1.53 against 1.34, so it restores the number and not the calibration, and AUTO-τ over-covers the same way. The gap between the best-calibrated temper and plain conformal is 0.050 under controlled covariate shift and 0.005 to 0.026 on real ACS shift, and on Wine the oracle lifts coverage from 0.70 only to 0.80–0.82 while 50 target labels reach 0.89–0.91 (Table 18).

A sharper ratio does not change this. Table 2 repeats the experiment with a cross-fitted gradient-boosted domain classifier, its isotonic-calibrated version, and TabPFNv2 itself as the domain classifier. The stronger estimators separate the domains more sharply (n_{eff} from 0.67 to 0.37), and a temper within 0.02 of nominal then exists in 76–83% of settings against 69%; but only test labels find it, its mean gain over plain conformal is 0.02 to 0.03, and the label-free AUTO-τ over-covers by 0.02 to 0.03 at 13% wider sets under every estimator. Nor is the residual label shift: weighting the calibration scores by the target-to-source label ratio (Podkopaev & Ramdas, 2021), estimated by black-box shift estimation or given the oracle target proportions, changes mean coverage by under

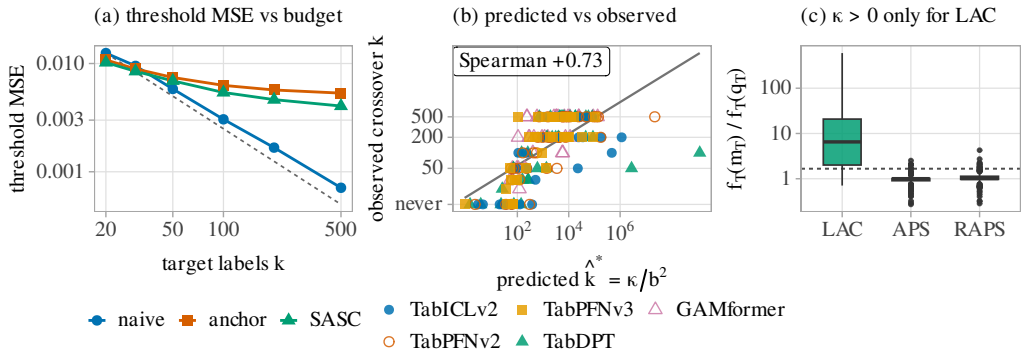


Figure 4: The crossover budget, tested three ways. (a) Threshold mean squared error against budget (TabICLv2); (b) predicted $\hat{k}^\star$ against the observed crossover, one point per setting and TFM; (c) the density ratio that sets the sign of κ, by score (dashed: break-even 1.67).

0.01 on both real-shift families (Table 6, Appendix D.5). Per pair, the target with the most covariate movement, the lowest n_{eff}, is not the one with the worst coverage (Figure 12).

Target labels are the fallback. Labeled target points are exchangeable with the test set, so split conformal on them is valid under any shift: on ACS, k labels drive coverage to nominal and lift the validity rate from 0.80 to 0.92 at k=500 (Table 21), most where shift is worst, and 50 labels give the same tail coverage as the best label-free temper (0.877 against 0.878) with 10% narrower sets. Small k fixes the mean but not the variance: at k=50 the validity rate is 0.68, so reliable coverage costs a few hundred labels (Appendix K). These results answer RQ2 in the negative: no label-free temper recalibrates real tabular shift, under four density-ratio estimators or label-shift weights, and whether the residual is a conditional shift or a misestimated ratio the experiment cannot tell.

Takeaway 2

No label-free reweighting rule recalibrates real tabular shift: sharper density ratios and label-shift weights move width, not calibration, and fifty target labels match the best temper's tail coverage with narrower sets.

4.3 RQ3: The crossover budget and the per-cell regime

To answer RQ3 we test Proposition 1 three ways, with densities estimated from the target test scores (Figure 4). First, against the true target quantile on the 46-setting suite, the naive threshold's mean squared error falls as $1/k$, the anchor's plateaus at its bias floor, and SASC tracks the lower envelope only for $k \leq 30$; beyond that the gate keeps the over-covering anchor, so SASC's threshold risk plateaus too, at 5.7× naive's by k=500. Second, the predicted $\hat{k}^\star$ orders the observed crossover with Spearman $\rho = +0.73$ pooled over the five TFMs and +0.66 to +0.84 per TFM; where $\kappa \leq 0$ the anchor never wins in 86–100% of settings for four TFMs and 46% for GAMformer. Third, LAC scores have $f_T(m_T)/f_T(q_T) \approx 2.5$ on ACS and 6–9 on OpenML, so $\kappa > 0$, whereas APS and RAPS have a ratio near 1 and anchoring fails under them as predicted (Appendix O). From the source scores and k labels alone the ordering is only partly recovered ($\rho = +0.36$ to $+0.69$) and the level not at all (Appendix U).

A larger bias shortens the regime, and the 11 UCI shifts test this. They are more severe than ACS: source-only calibration attains a tail coverage of 0.67–0.69 on the nine tabular settings and almost nothing on the two sensor-drift ones, and the anchor bias is ten times larger (median $|b| = 0.24$ against 0.02), so (4) gives a small $k^\star$. That is what we observe (Table 42, Appendix S): naive few-shot is already reliable at k=50, both gated rules match it at a 5–9% size cost, SASC sits below naive at k=20 (0.854 against 0.892 for TabICLv2), and WCC-CSPD collapses to 0.44–0.48.

The regime the budget locates is few labels per calibration cell, and conditional anchoring is the test of whether it is usable (Figure 5). Marginal calibration hides a conditional deficit: on the worst of three confidence bins, naive few-shot covers 0.75–0.77 and marginal SASC 0.77–0.80 at k=50

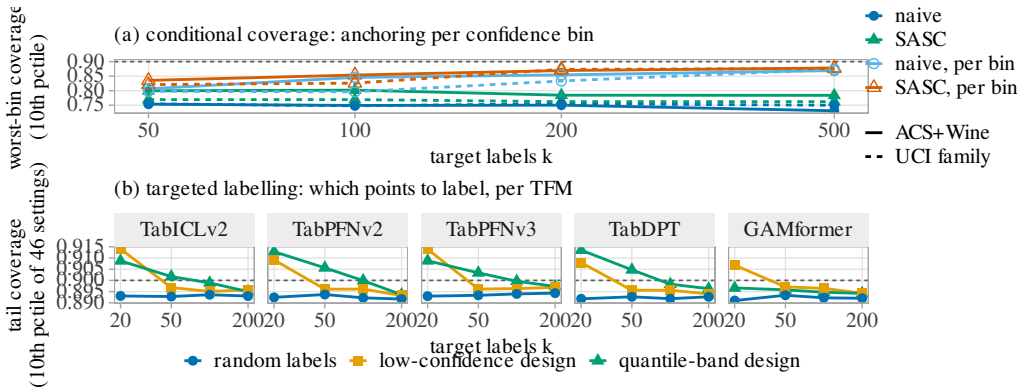


Figure 5: Two ways to spend the label budget. (a) Worst-bin coverage against budget (TabICLv2; ACS solid, UCI dashed). (b) Tail coverage of naive few-shot with random and quantile-band labels, per TFM.

on ACS, and less on the UCI family. Calibrating per bin lifts this to 0.81–0.82, and anchoring per bin to 0.835–0.846, the level per-bin naive reaches only at k=100–200. Against per-bin target-only calibration at matched width the point estimates stay positive at k=50, +0.024 to +0.029 for four of five TFMs, but the paired intervals exclude zero only for TabPFNv3 and TabDPT (Appendix X.1). Labeling near the source quantile instead of at random, with design weights that keep the guarantee exact, raises naive few-shot's tail coverage by 0.01 to 0.02 at every budget for all five TFMs, about what the anchor buys, and the two add only partially (Appendix Y). These results answer RQ3: the budget orders where the anchor has lower risk and explains its failure on cumulative-mass scores and on the severe family, but it is retrospective, and the per-cell gain it points to is suggestive rather than established.

Takeaway 3

The crossover budget says where the anchor can pay, after the fact: LAC scores and mild shifts give a short regime at small k, cumulative-mass scores and severe shifts give none.

All of the above holds beyond TFMs: every result holds for the five TFMs to within 0.01 and for XGBoost and logistic regression (Appendix T), GAMformer, whose additive scores keep their median-to-tail spread under shift, has the smallest anchor bias (median $|b| = 0.017$ against 0.024) and the only gate that beats naive at every budget, and regression behaves the same way under three score functions (Appendix V). Three natural alternatives fail (Appendix Y.6): embedding-space density ratios, resampling the in-context support toward the target, and temperature scaling (Guo et al., 2017), which cannot move coverage.

5 Conclusion

Reusing the source calibration set changes what a coverage number estimates. A rule that pools or anchors to the source returns a different width than target-only calibration, so its tail coverage at the nominal level mixes calibration with conservatism. The width-matched control separates the two by reading target-only calibration at an inflated rank at the rule's own set size, per calibration draw, with paired cluster bootstraps over settings and tasks. For source-anchored rules, the crossover budget $k^\star = \kappa/b^2$ says when the anchor has lower risk, retrospectively. Experiments on five TFMs and 57 real shifts show that source-reuse gains are mostly width, that the gated rules keep a small per-draw gain only at the smallest budgets, and that no label-free reweighting rule recalibrates real shift. In practice, if importance-weighted and unweighted coverage agree on a held-out fold yet coverage is low, reweighting will not help; collect a few dozen target labels and inflate the target-only rank, gating the source only at the smallest budgets. Identifying the residual shift directly remains open.

Reproducibility statement

Every number in the paper is produced from public data by the scripts and cached results at `https://anonymous.4open.science/r/TFM-1-ICLR2027-576E/`, which regenerate every table and figure on a CPU in minutes and every study from the cached predictions in a few hours. Appendix E lists the datasets (OpenML ids, ACS/folktables task definitions, Wine) and preprocessing; Appendix F gives the protocol, seeds, hardware, runtimes and a reproducibility checklist; Appendix C states SASC and AUTO-τ in full; Appendix D contains all proofs. All appendix studies are deterministic post-processing of one cached set of base-model predictions, so they can be re-run in minutes without a GPU.

Ethics statement

This work uses only public, de-identified data and frozen public models. Its aim is to make the uncertainty that deployed tabular models report more reliable under shift. Appendix Z discusses the ethics of acquiring target labels, the distinction between marginal coverage and group fairness, and the conditions under which the certified variant, rather than the heuristic, should be used.

Use of large language models

During the preparation of this paper, we used a large language model to draft and revise text and to improve its grammar, word choice, and conciseness. We also used it to help identify relevant literature. The authors independently read all suggested papers and verified each citation against its original source before inclusion.

We used Claude Code to assist with designing and running the experiments and analyzing the results. The authors reviewed the code and verified all reported numerical results against the saved outputs. All claims are the authors' own, and we take full responsibility for the content of this paper.

References

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Anastasios N Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://iclr.cc/virtual/2021/poster/3246`.

Anastasios N Angelopoulos, Emmanuel J Candès, and Ryan J Tibshirani. Conformal PID control for time series prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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. doi: 10.1214/24-AOAS1998. URL `https://doi.org/10.1214/24-AOAS1998`.

Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021a.

Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10(2):455–482, 2021b.

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.

Hamsa Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5):2964–2984, 2021.

Aadyot Bhatnagar, Huan Wang, Caiming Xiong, and Yu Bai. Improved online conformal prediction via strongly adaptive online learning. In *International Conference on Machine Learning (ICML)*, 2023. URL `https://proceedings.mlr.press/v202/bhatnagar23a.html`.

Michael Bian and Rina Foygel Barber. Training-conditional coverage for distribution-free predictive inference. *Electronic Journal of Statistics*, 17(2):2044–2066, 2023.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137–2155, 2009.

T Tony Cai and Hongji Wei. Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *The Annals of Statistics*, 49(1):100–128, 2021.

Maxime Cauchois, Suyash Gupta, Alnur Ali, and John C Duchi. Robust validation: Confident predictions even when distributions shift. *Journal of the American Statistical Association*, 119(548): 3033–3044, 2024.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016. doi: 10.1145/2939672.2939785.

Alvaro H. C. Correia and Christos Louizos. Non-exchangeable conformal prediction with optimal transport: Tackling distribution shift with unlabeled data. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 175380–175416, 2025. doi: 10.52202/085713-5278. URL https://doi.org/10.52202/085713-5278. arXiv:2507.10425.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/32e54441e6382a7fbacbbbaf3c450059-Abstract.html.

Bradley Efron and Carl Morris. Data analysis using Stein's estimator and its generalizations. *Journal of the American Statistical Association*, 70(350):311–319, 1975.

Clara Fannjiang, Stephen Bates, Anastasios N Angelopoulos, Jennifer Listgarten, and Michael I Jordan. Conformal prediction under feedback covariate shift for biomolecular design. *Proceedings of the National Academy of Sciences*, 119(43), 2022.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with TableShift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html.

Isaac Gibbs and Emmanuel J Candès. Conformal inference for online prediction with arbitrary distribution shifts. *Journal of Machine Learning Research*, 25(162):1–36, 2024. URL https://jmlr.org/papers/v25/22-1218.html.

Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(4):1100–1126, 2025. doi: 10.1093/jrsssb/qkaf008. URL https://doi.org/10.1093/jrsssb/qkaf008. arXiv:2305.12616.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks*, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. TabPFN-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026. URL https://arxiv.org/abs/2605.13986.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://iclr.cc/virtual/2023/poster/12113.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 361–379. University of California Press, 1961. URL https://projecteuclid.org/euclid.bsmsp/1200512173.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

Byol Kim, Chen Xu, and Rina Foygel Barber. Predictive inference is free with the jackknife+-after-bootstrap. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Sai Li, T Tony Cai, and Hongzhe Li. Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B*, 84 (1):149–173, 2022.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 191387–191417, 2025. doi: 10.52202/085713-5748. URL https://doi.org/10.52202/085713-5748. arXiv:2410.18164.

Pascal Massart. The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks*, 2023.

Yinjie Min, Liuhua Peng, and Changliang Zou. Stable localized conformal prediction via transduction. *arXiv preprint arXiv:2605.01452*, 2026. URL https://arxiv.org/abs/2605.01452.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodriguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

Andreas Mueller, Julien Siems, Harsha Nori, David Salinas, Arber Zela, Rich Caruana, and Frank Hutter. GAMformer: Bridging tabular foundation models and interpretable machine learning. *arXiv preprint arXiv:2410.04560*, 2024. URL https://arxiv.org/abs/2410.04560.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://iclr.cc/virtual/2022/poster/6595.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning (ICML)*, 2023. URL https://proceedings.mlr.press/v202/nagler23a.html.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning (ECML)*, 2002. doi: 10.1007/3-540-36755-1_29.

Sangdon Park, Edgar Dobriban, Insup Lee, and Osbert Bastani. PAC prediction sets under covariate shift. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://iclr.cc/virtual/2022/poster/6314.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in Artificial Intelligence (UAI)*, 2021. URL https://proceedings.mlr.press/v161/podkopaev21a.html.

Hongxiang Qiu, Edgar Dobriban, and Eric Tchetgen Tchetgen. Prediction sets adaptive to unknown covariate shift. *Journal of the Royal Statistical Society Series B*, 85(5):1680–1705, 2023.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/qu25d.html.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026. URL https://arxiv.org/abs/2602.11139.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/5103c3584b063c431bd1268e9b5e76fb-Abstract.html.

Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/244edd7e85dc81602b7615cd705545f5-Abstract.html.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Robert J Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley, 1980.

Matteo Sesia and Emmanuel J Candès. A comparison of some conformal quantile regression methods. *Stat*, 9(1), 2020.

Farbod Siahkali, Ashwin Verma, and Vijay Gupta. Coverage guarantees for pseudo-calibrated conformal prediction under distribution shift. *arXiv preprint arXiv:2602.14913*, 2026. URL https://arxiv.org/abs/2602.14913.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract.html.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html.

Joaquin Vanschoren, Jan N van Rijn, Bernd Bischl, and Luis Torgo. OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014. doi: 10.1145/2641190.2641198.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning (ACML)*, 2012. URL https://proceedings.mlr.press/v25/vovk12.html.

Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74: 9–28, 2015.

Vladimir Vovk and Ivan Petej. Venn–Abers predictors. In *Uncertainty in Artificial Intelligence (UAI)*, 2014. URL https://www.auai.org/uai2014/proceedings/individuals/166.pdf.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Baozhen Wang and Xingye Qiao. Conformal prediction under generalized covariate shift with posterior drift. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, 2025. URL https://proceedings.mlr.press/v258/wang25l.html.

Yachong Yang, Arun Kumar Kuchibhotla, and Eric Tchetgen Tchetgen. Doubly robust calibration of prediction sets under covariate shift. *Journal of the Royal Statistical Society Series B*, 86(4): 943–965, 2024.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. Adaptive conformal predictions for time series. In *International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/zaffran22a.html.

Appendix overview

The appendix is organized as follows. §A fixes notation and the shift taxonomy. §B gives the related work. §C states every algorithm in full, with complexity. §D gives all proofs and two results not in the main text: a bias–variance characterization of the source anchor (§D.3) that predicts *when* SASC helps, and a lower bound on SASC's coverage in terms of the anchor bias (§D.4). §E–§F document data, protocol, compute and reproducibility. §G–§K give the per-setting tables behind the main text's averages, including the 50k-scale experiment, the AUTO-τ algorithm and controlled-shift results. §L–§X are the new studies, all computed from one cached set of base-model predictions over 46 settings, 5 seeds and five base models. They cover the full per-setting breakdown, coverage-level and score-function sweeps, SASC's internals, the conservative form's δ, and source calibration size. They also cover a second real-shift family from public UCI tables (§S), non-foundation base models, an anatomy of the score-distribution shift with a direct test of the theory, regression with CQR and normalized scores, class-conditional composition, and significance testing. §U.1–§Z test whether the density-ratio weights are informative, give the negative results in full, and discuss broader impact.

A Notation, formal setup, and the shift taxonomy

A.1 Conformal prediction on a frozen predictor

Let $\mathcal{X}$ be the covariate space and $\mathcal{Y}$ the label space ($\mathcal{Y} = \{1, \dots, C\}$ for classification, $\mathcal{Y} = \mathbb{R}$ for regression). We use a tabular foundation model as a frozen map $f : \mathcal{X} \to \Delta^{C-1}$ (or $\mathcal{X} \to \mathbb{R}$), obtained by conditioning on an in-context support set $\mathcal{D}_{\text{sup}} = \{(x_i^{\text{sup}}, y_i^{\text{sup}})\}_{i=1}^{n_{\text{tr}}}$ drawn from the *source* distribution P_S. Thereafter f is fixed; all randomness is over the calibration and test draws, conditional on f.

A *nonconformity score* $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ measures disagreement between $f(x)$ and a candidate label. We use

$$\text{LAC} \quad s(x,y) = 1 - \hat{p}_y(x), \tag{5}$$

$$\text{APS} \quad s(x,y) = \textstyle\sum_{y' \,:\, \hat{p}_{y'}(x) \geq \hat{p}_y(x)} \hat{p}_{y'}(x), \tag{6}$$

$$\text{RAPS} \quad s(x,y) = \text{APS}(x,y) + \lambda \left(\text{rank}(y \mid x) - k_{\text{reg}}\right)_+, \tag{7}$$

$$\text{absolute residual} \quad s(x,y) = |y - \hat{y}(x)|, \tag{8}$$

$$\text{normalized residual} \quad s(x,y) = |y - \hat{y}(x)| / \hat{\sigma}(x), \tag{9}$$

$$\text{CQR} \quad s(x,y) = \max\{\hat{q}_{\alpha/2}(x) - y,\ y - \hat{q}_{1-\alpha/2}(x)\}, \tag{10}$$

where $\text{rank}(y \mid x)$ is the position of y when classes are sorted by decreasing $\hat{p}$. LAC (Vovk et al., 2005) and the absolute residual (Papadopoulos et al., 2002; Lei et al., 2018) are the main-text choices. APS (Romano et al., 2020), RAPS (Angelopoulos et al., 2021), the normalized residual and CQR (Romano et al., 2019) appear in the score ablations (§O, §V).

Given calibration scores $s_1, \ldots, s_n$ and a level α, split conformal (Papadopoulos et al., 2002; Lei et al., 2018) thresholds at the empirical quantile

$$\hat{q} \;=\; \hat{Q}_{1-\alpha}(\{s_i\}) \;=\; s_{(\lceil (n+1)(1-\alpha) \rceil)}, \tag{11}$$

($= +\infty$ when $\lceil (n+1)(1-\alpha) \rceil > n$) and outputs $C(x) = \{y : s(x,y) \leq \hat{q}\}$. If $(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, y_{n+1})$ are exchangeable, then $\Pr(y_{n+1} \in C(x_{n+1})) \geq 1 - \alpha$, with the complementary upper bound $1 - \alpha + 1/(n+1)$ when scores are almost surely distinct.

Two notions of validity recur below.

- **Marginal coverage:** $\Pr(Y \in C(X)) \geq 1 - \alpha$, where the probability is over *both* the calibration draw and the test point. This is what split conformal delivers.
- **Training-conditional coverage:** $\Pr_{\text{cal}}\big(\Pr_{\text{test}}(Y \in C(X) \mid \text{cal}) \geq 1 - \alpha\big) \geq 1 - \delta$, i.e. the realized coverage of the *deployed* predictor is at least $1 - \alpha$ for all but a δ fraction of calibration draws (Vovk, 2012; Bian & Barber, 2023). A practitioner who calibrates once and then deploys wants this guarantee, and Prop. 3 provides it.

Marginal validity says nothing about the realized coverage of any *particular* calibration set. With $k = 50$ target labels, split conformal is marginally valid, yet its realized coverage has a standard deviation of about $\sqrt{\alpha(1-\alpha)/k} \approx 4$ percentage points. This is what "unreliable at small k" means throughout the paper: variance, not bias.

A.2 Shift taxonomy

Write P_S and P_T for the source and target joint laws, each factorized as $P(X)P(Y \mid X)$. Following Quiñonero-Candela et al. (2009); Moreno-Torres et al. (2012):

- **Covariate shift:** $P_S(Y \mid X) = P_T(Y \mid X)$ and $P_S(X) \neq P_T(X)$. The density ratio $w(x) = \mathrm{d}P_T(x)/\mathrm{d}P_S(x)$ suffices to transport any expectation; weighted conformal (Tibshirani et al., 2019) uses this.
- **Concept shift:** $P_S(Y \mid X) \neq P_T(Y \mid X)$. No function of X alone can correct it, because the label relationship itself has changed.
- **Label shift:** $P_S(Y) \neq P_T(Y)$ with $P_S(X \mid Y) = P_T(X \mid Y)$ (Saerens et al., 2002; Lipton et al., 2018; Podkopaev & Ramdas, 2021). This third regime is correctable without target labels given a well-specified generative direction. We do not assume it, and the ACS shifts we study do not satisfy it, since both the covariate marginal and the conditional move.

Real shifts are mixtures. Our empirical claim is narrower than attributing the conformal failure to one component: it is that the evaluated covariate reweighting removes little of the coverage deficit on real tabular shift, since the best member of the family removes little, and that we could not determine whether the remainder is a conditional change or an unestimable ratio (§D.5).

A.3 Quantities used throughout

For density-ratio weights $w_1, \ldots, w_n$ on calibration points, the *effective sample size* and its normalized version are

$$n_{\text{eff}} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2}, \qquad n_{\text{eff}}^{\text{frac}} = n_{\text{eff}}/n \in (0, 1]. \tag{12}$$

$n_{\text{eff}}^{\text{frac}}$ measures how much the weights concentrate: it is 1 under no shift and tends to 0 when a few calibration points carry all the mass. It controls the *variance* of any importance-weighted estimate, hence how noisy weighted conformal is, but says nothing about whether reweighting can remove the coverage deficit at all (§4.2 shows this empirically, §D.5 explains it).

The *oracle gap* is

$$\Delta \;=\; \max_{\tau \in \mathcal{T}} \text{cov}(\tau) \;-\; \text{cov}(0), \tag{13}$$

where $\text{cov}(\tau)$ is the realized target coverage of tempered weighted conformal with weights w^τ (so $\tau = 0$ is plain split conformal and $\tau = 1$ is fully weighted conformal). We take the maximum using *test labels*, so Δ is an oracle upper bound on what any selection rule inside the temper family can achieve.

We report five quantities. The first two are coverage and *coverage error* $|\text{cov} - (1-\alpha)|$. The third is *tail coverage*, the 10th percentile of the per-setting coverage distribution, i.e. the reliability of the hardest-to-cover tenth of deployments. The fourth is the *validity rate*, the fraction of settings with $\text{cov} \geq 1 - \alpha - 0.02$. The fifth is average set size for classification, or interval width in target standard deviations for regression. For class-conditional analyses we also report *worst-class coverage*, $\min_c \Pr(Y \in C(X) \mid Y = c)$.

B Extended related work

Conformal prediction. Conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023) turns any point predictor into set predictions with finite-sample marginal coverage under exchangeability. The split (inductive) variant (Papadopoulos et al., 2002; Lei et al., 2018) is the practical workhorse and the right choice here: tabular foundation models are expensive to re-condition, so cross-conformal (Vovk, 2015), jackknife+ (Barber et al., 2021a) and jackknife+-after-bootstrap (Kim et al., 2020), which need many refits, are less attractive. In-context learning does make leave-one-out cheap, and we regard conformalizing the ICL support set as an open direction. Score design is largely orthogonal to our contribution. LAC gives the smallest sets, APS (Romano et al., 2020) and RAPS (Angelopoulos et al., 2021) trade size for better conditional coverage, and CQR (Romano et al., 2019; Sesia & Candès, 2020) gives length-adaptive regression intervals. §O and §V show that our conclusions do not depend on this choice, as one expects of a statement about *which distribution the calibration scores come from* rather than *how the scores are formed*. Conditional coverage beyond the marginal guarantee is impossible in a distribution-free sense (Barber et al., 2021b; Vovk, 2012). Mondrian/class-conditional calibration (Vovk, 2012) buys group-wise validity at a variance cost; we quantify this trade in §4.2 and §W.1.

Conformal prediction under distribution shift. Weighted conformal (Tibshirani et al., 2019) restores coverage under covariate shift with a known density ratio. A large family of methods builds on it: PAC prediction sets under covariate shift (Park et al., 2022), shift-adaptive sets when the ratio is unknown (Qiu et al., 2023), doubly robust calibration (Yang et al., 2024), distributionally robust validity over f-divergence balls (Cauchois et al., 2024), and feedback covariate shift in design loops (Fannjiang et al., 2022). All of these inherit the covariate-shift assumption $P_S(Y \mid X) = P_T(Y \mid X)$. Our reality check measures how much that assumption is worth on real tabular shift; the answer (§4.2) is little, on the shifts that dominate tabular deployment. The density ratio itself can be estimated by a domain classifier (Bickel et al., 2009) or by direct ratio estimation (Sugiyama et al., 2007; Kanamori et al., 2009); §U.1 shows our conclusions are not an artifact of the estimator, since even the *oracle* temper within the family fails. A second family drops i.i.d. assumptions entirely and adapts online: adaptive conformal inference (Gibbs & Candès, 2021; 2024), adaptive conformal for time series (Zaffran et al., 2022), strongly adaptive online conformal (Bhatnagar et al., 2023), and conformal PID control (Angelopoulos et al., 2023). These need a *stream* of labeled target feedback,

a strictly stronger requirement than the one-off labeled batch we assume; when such a stream exists they are the right tool. Finally, Barber et al. (2023) bound the coverage gap under non-exchangeability by a total-variation term. The term is unknown, so the bound is informative but cannot be acted on by itself. Our oracle-gap and n_{eff} pair is a cheap empirical proxy for *which* correction can close a given gap.

Label shift and post-hoc calibration. When only $P(Y)$ moves, EM-style prior correction (Saerens et al., 2002) and black-box shift estimation (Lipton et al., 2018) apply, and Podkopaev & Ramdas (2021) give distribution-free sets under label shift. Post-hoc probability calibration, such as Platt scaling (Platt, 1999), isotonic regression (Zadrozny & Elkan, 2002) and temperature scaling (Guo et al., 2017), improves the reliability of probabilities but does nothing for conformal coverage under shift (§4.3, §Y.6), because the conformal quantile absorbs any monotone rescaling of $\hat{p}$. Venn–Abers predictors (Vovk & Petej, 2014) give calibrated probabilities with validity guarantees, but again under exchangeability, so they do not address the failure mode studied here.

Tabular foundation models. Prior-data-fitted networks (Müller et al., 2022; Hollmann et al., 2023; 2025) approximate posterior predictive inference by amortized training on a synthetic prior; Nagler (2023) gives statistical foundations. TabICL (Qu et al., 2025; 2026) scales in-context tabular learning to large tables with column-then-row attention. These models now match or beat tuned gradient boosting on public benchmarks, reversing a long-standing gap (Grinsztajn et al., 2022; McElfresh et al., 2023). Their reliability under shift is largely unmeasured, and shift is where the amortized-prior story is least obviously safe. The synthetic prior is a distribution over *functions and covariates*, so nothing in the pretraining objective anticipates a target whose $P(Y \mid X)$ differs from the source's. Our diagnosis (§4) confirms the concern for both model families. §T shows that gradient boosting and logistic regression lose coverage on the same settings, so the phenomenon is not specific to in-context learning and the prescription is about the shift, not the architecture.

Shrinkage, transfer, and where SASC sits. SASC is a shrinkage estimator in the tradition of James & Stein (1961); Efron & Morris (1975). It trades a controlled bias (the source anchor) for a large variance reduction, and the best trade depends on the size of the bias relative to the noise (§D.3 makes this exact for the conformal quantile). It is also a transfer-learning estimator in the sense of Bastani (2021); Li et al. (2022); Cai & Wei (2021), where abundant source data reduce the variance of a scarce-target estimate at a penalty set by how different the two are. What is new here is not shrinkage itself but *what is shrunk*: the *conformal quantile*, the one statistic that determines coverage, rather than the predictor or the whole score distribution. We also select the shrinkage weight by coverage-targeted cross-validation on the target labels, which makes the estimator self-limiting: it reverts to the exactly-valid target-only quantile whenever the anchor fails its own check. To our knowledge, this combination (source-anchored quantile shrinkage with a coverage-gated weight, plus a training-conditional floor for a certified variant) has not been used for conformal calibration under concept shift.

Positioning. The two closest lines are (i) conformal prediction for prior-data-fitted networks, studied in-distribution and with missing data, and (ii) covariate-shift conformal, classically evaluated on synthetic or semi-synthetic covariate shift, where it works by construction. Relative to (i), we stress-test under *real* distribution shift and report a negative result specific to that regime. Relative to (ii), we measure what fraction of real tabular shift the method can address, attribute the remainder with a cheap diagnostic, and give a label-efficient alternative for the part that remains. We include two base models, two task types, a non-ACS benchmark and non-foundation controls so that the conclusion cannot be dismissed as model- or dataset-specific.

Closest prior methods and this paper. Wang & Qiao (2025) assume covariate shift with posterior drift and pool source and labeled target points in a class-wise weighted conformal classifier with class-conditional density-ratio weights; on our two real-shift families it falls below target-only calibration (§4.1), as do methods that use unlabeled target data only (Correia & Louizos, 2025; Siahkali et al., 2026; Min et al., 2026). Label-shift conformal (Podkopaev & Ramdas, 2021) is our baseline for ruling out label shift (§D.7). Gibbs et al. (2025) give guarantees that hold simultaneously over a class of covariate shifts, the conditional validity that §4.3 measures with confidence bins. Conformal prediction for prior-data-fitted and in-context tabular models (Müller et al., 2022; Hollmann et al.,

2023; 2025; Qu et al., 2025; 2026) is valid in distribution and, we show, breaks under real shift. SASC is a shrinkage estimator (James & Stein, 1961; Bastani, 2021; Li et al., 2022) with a coverage-gated weight; what this paper adds is not the estimator but the control against which any such estimator should be read.

C ALGORITHMS IN FULL

Algorithm 1 states SASC exactly as run, including the inner cross-validation and the conservative floor; Algorithm 2 in §I does the same for AUTO-τ.

Algorithm 1 SASC: source-anchored shrinkage conformal calibration

Input: source calibration scores $\{s_i\}_{i=1}^{n_s}$; target scores $\{t_j\}_{j=1}^{k}$; level α; folds F ($= 5$); weight grid $\mathcal{W}$ ($= \{0, .1, .25, .4, .55, .7, .85, 1\}$); mode $\in \{\text{point}, \text{conservative}\}$; confidence δ
$q_{\text{src}} \leftarrow \hat{Q}_{1-\alpha}(\{s_i\})$; $\quad m_{\text{src}} \leftarrow \text{med}(\{s_i\})$ ▷ low-variance source shape, $n_s \gg k$
if $k < 2F$ **then** $w^\star \leftarrow 1$ **else**
  partition $\{1..k\}$ into folds $\mathcal{F}_1, \ldots, \mathcal{F}_F$ (fixed permutation, seed 0)
  for $w \in \mathcal{W}$ *ascending* (most anchor use first)
    for $f = 1..F$: $q^{(f)} \leftarrow w\,\hat{Q}_{1-\alpha}\big(\{t_j\}_{j\notin\mathcal{F}_f}\big) + (1-w)\big[q_{\text{src}} + \text{med}(\{t_j\}_{j\notin\mathcal{F}_f}) - m_{\text{src}}\big]$
    $\widehat{\text{cov}}(w) \leftarrow \frac{1}{F}\sum_f \frac{1}{|\mathcal{F}_f|}\sum_{j\in\mathcal{F}_f} \mathbf{1}[t_j \le q^{(f)}]$ ▷ held-out target coverage
    if $\widehat{\text{cov}}(w) \ge 1-\alpha$ **then** $w^\star \leftarrow w$; **break**
  if no w passed **then** $w^\star \leftarrow 1$ ▷ fall back to exactly-valid naive
$\hat{q} \leftarrow w^\star\,\hat{Q}_{1-\alpha}(\{t_j\}) + (1-w^\star)\big[q_{\text{src}} + \text{med}(\{t_j\}) - m_{\text{src}}\big]$
if mode $=$ conservative **then** $\hat{q} \leftarrow \max\{\hat{q},\ t_{(r)}\}$ with $r = \min\{r : 1 - \text{Beta}_{r,k-r+1}(1-\alpha) \ge 1-\delta\}$
return $\hat{q}$; deploy $C(x) = \{y : s(x,y) \le \hat{q}\}$

Complexity. Sorting the source scores costs $O(n_s \log n_s)$ once; we reuse the result across all k and all candidate weights. Each candidate weight costs $O(F\,k \log k)$, so the inner selection is $O(|\mathcal{W}|Fk \log k)$; with $k = 500$, $F = 5$, $|\mathcal{W}| = 8$ this is under a millisecond. SASC adds no base-model forward passes over naive few-shot conformal, since we compute the target scores once, so its runtime overhead is negligible next to a single TFM inference pass (§F, Table 10). Memory is $O(n_s + k)$.

Design choices and why they are what they are. Three details matter. *(i) Ascending weight order.* We scan w from 0 (pure anchor) upward and stop at the first weight that passes, i.e. we use as much of the anchor as the data permit. Scanning downward from 1 to the last passing weight would be equivalent only if $\widehat{\text{cov}}(w)$ were monotone in w, which it is not in finite samples. *(ii) The fallback is naive, not the anchor.* When *no* weight passes we return $w^\star = 1$, the exactly-valid target-only quantile. We stress what this does and does not buy. It removes the failure mode in which an empty candidate set leaves the rule undefined, but it does *not* make the point form valid: the gate can still accept a weight whose held-out coverage estimate exceeded $1-\alpha$ by chance while its true coverage is below it, and that event is not controlled. The point form is a heuristic, as §3.2 says, and only the conservative floor carries a guarantee. *(iii) The median, not the mean, defines the anchor's location.* LAC scores are bounded in $[0, 1]$ but residuals are heavy-tailed, so the median gap is the safer choice; §P confirms that it beats the mean and 75th-percentile variants, while the two-parameter location–scale anchor over-corrects.

D THEORY AND PROOFS

This section gives the two proofs the main text refers to (§D.1, §D.2) and three new results. §D.3 characterizes the source anchor's bias and variance, which predicts *when* SASC helps and at what label budget the benefit expires. §D.4 lower-bounds SASC's realized coverage in terms of the anchor bias, with high probability, and explains why the point form has no distribution-free guarantee. §D.5 formalizes the oracle-gap diagnostic.

Throughout, scores are real-valued with continuous distribution functions F_S (source calibration) and F_T (target), with densities f_S, f_T that are continuous and strictly positive near the quantiles used. Write

$$q_S = F_S^{-1}(1-\alpha), \quad m_S = F_S^{-1}(1/2), \quad q_T = F_T^{-1}(1-\alpha), \quad m_T = F_T^{-1}(1/2). \tag{14}$$

The k target calibration scores $t_1, \ldots, t_k$ are i.i.d. F_T and exchangeable with the test score, and the n_s source scores are i.i.d. F_S with $n_s \gg k$.

D.1 Proof of Proposition 7 (auto-τ selection criterion)

Under covariate shift, $P_S(Y \mid X) = P_T(Y \mid X)$ and $w = \mathrm{d}P_T/\mathrm{d}P_S$ on $\mathcal{X}$. For any integrable g,

$$\mathbb{E}_T[g(X,Y)] = \iint g \,\mathrm{d}P_T(y \mid x)\,\mathrm{d}P_T(x) = \iint g \,\mathrm{d}P_S(y \mid x)\, w(x)\,\mathrm{d}P_S(x) = \mathbb{E}_S[w\, g], \tag{15}$$

where the middle equality uses *both* halves of the covariate-shift assumption: the conditional is shared, and the marginals differ by w. Taking $g(x,y) = \mathbf{1}[y \in C(x)]$ for a fixed set-valued map C (measurable and independent of the selection fold, which is why auto-τ splits calibration into FIT and SEL) gives $\mathrm{cov}_T(C) = \mathbb{E}_S[w\,\mathbf{1}(Y \in C(X))]$. The self-normalized estimator on the selection fold,

$$\hat{\mu} = \frac{\sum_{j \in \mathrm{SEL}} w_j\, \mathbf{1}[Y_j \in C(X_j)]}{\sum_{j \in \mathrm{SEL}} w_j}, \tag{16}$$

is a ratio of two averages that converge a.s. to $\mathbb{E}_S[w\mathbf{1}(\cdot)]$ and $\mathbb{E}_S[w]$. By the strong law and the continuous mapping theorem, $\hat{\mu} \to \mathrm{cov}_T(C)$ a.s. as $|\mathrm{SEL}| \to \infty$, provided $\mathbb{E}_S[w] < \infty$. Self-normalization also removes any unknown positive constant in w, which matters because a domain classifier identifies w only up to the class-prior ratio.

For the variance, the delta method for a self-normalized importance-sampling ratio gives $\mathrm{Var}(\hat{\mu}) \approx \mathbb{E}_S[w^2(\mathbf{1} - \mu)^2]/(n\,\mathbb{E}_S[w]^2)$. For a bounded Bernoulli integrand this is at most $\mu(1-\mu)\,\mathbb{E}_S[w^2]/(n\mathbb{E}_S[w]^2) = \mu(1-\mu)/n_{\mathrm{eff}}$ with n_{eff} as in Eq. (12). This motivates $\widehat{\mathrm{se}} = \sqrt{\hat{\mu}(1-\hat{\mu})/n_{\mathrm{eff}}}$ and the conservative rule $\hat{\mu} - z\widehat{\mathrm{se}} \geq 1 - \alpha$. The multiplier z turns a noisy point estimate into an approximate lower confidence bound; its effect is exactly the validity/width knob we ablate in §I.

Finally, if $P_S(Y \mid X) \neq P_T(Y \mid X)$ then the second equality in Eq. (15) fails, and re-weighting does not repair the failure. Proposition 2 below makes this precise. ■

Proposition 2 (Reweighting is blind to concept shift)**.** *Fix any set-valued map C and any non-negative weight function u on $\mathcal{X}$ with $\mathbb{E}_S[u] = 1$. Define the u-reweighted source coverage $\mathrm{cov}_u(C) = \mathbb{E}_S[u(X)\mathbf{1}(Y \in C(X))]$. Then*

$$\mathrm{cov}_T(C) - \mathrm{cov}_u(C) = \underbrace{\int c_S(x)\big[\mathrm{d}P_T(x) - u(x)\mathrm{d}P_S(x)\big]}_{\textit{covariate term}} + \underbrace{\int \big[c_T(x) - c_S(x)\big]\mathrm{d}P_T(x)}_{\textit{concept term}}, \tag{17}$$

where $c_P(x) = P(Y \in C(x) \mid X = x)$. The covariate term vanishes at $u = w$; the concept term does not depend on u at all. No choice of weights can remove the concept term, and that includes any temper τ and any oracle within the family.

Proof. Add and subtract $\int c_S(x)\mathrm{d}P_T(x)$ and use $\mathrm{cov}_u(C) = \int c_S(x)u(x)\mathrm{d}P_S(x)$ and $\mathrm{cov}_T(C) = \int c_T(x)\mathrm{d}P_T(x)$. The concept term is a functional of $(c_T - c_S)$ and P_T only. □

The proposition says what covariate weights *could* never remove, for a fixed set: a genuine change in $c_T - c_S$. It does not say that the ACS residual is that term. Two things stand between the decomposition and that conclusion. The covariate term vanishes only at the true ratio $u = w$, and ours is estimated, so an unremoved residual can equally be a misestimated ratio; and the set C itself changes with the weights, so the two terms can offset. §D.5 reports three attempts to identify the concept term directly, all of which failed their own control. The supported statement is the one in §4.2: the evaluated reweighting procedures do not recalibrate on these shifts.

D.2 Proof of Proposition 3 (conservative validity)

Point and conservative forms. We use SASC in two forms. The *point* form deploys $\hat{q}_{\text{SASC}}$ with the cross-validated w. It is efficient, but selecting w on the target labels breaks exchangeability, so it is a heuristic with no finite-sample guarantee. The *conservative* form floors the threshold at a training-conditional target quantile, $\hat{q}^{\text{cons}}_{\text{SASC}} = \hat{q}_{\text{SASC}} \vee Q^{\text{lcb}}$, and is valid at a set-size cost that we report.

Proposition 3 (Conservative validity)**.** *Let Q^{lcb} be the target order statistic of rank $r(k, \alpha, \delta)$, the smallest r with $1 - \text{Beta}_{r,\, k-r+1}(1-\alpha) \geq 1 - \delta$. Then $\hat{q}^{\text{cons}}_{\text{SASC}} = \hat{q}_{\text{SASC}} \vee Q^{\text{lcb}}$ has training-conditional coverage $\Pr_{\text{cal}}\big(\Pr_{\text{test}}(Y \in C(X)) \geq 1-\alpha\big) \geq 1-\delta$, because its threshold is at least Q^{lcb} and raising a conformal threshold cannot lower coverage. As $k \to \infty$ both forms converge to the naive quantile only when the anchor's limiting bias satisfies $b \leq 0$; when $b > 0$ the gate accepts the anchor with probability tending to one and the limiting threshold is $q_T + b$, which over-covers. Limiting coverage is therefore at least $1 - \alpha$ in both cases, but it is exact only when $b \leq 0$ (Proof and empirics: Appendix D.2).*

Proof. Let $t_{(1)} \leq \cdots \leq t_{(k)}$ be the ordered target calibration scores and let $C_r(x) = \{y : s(x,y) \leq t_{(r)}\}$. Conditional on the calibration sample, the realized target coverage of C_r is $\text{cov}(r) = F_T(t_{(r)})$. Since F_T is continuous, $U_i := F_T(t_i)$ are i.i.d. Unif$(0,1)$, so $F_T(t_{(r)}) = U_{(r)} \sim \text{Beta}(r, k-r+1)$ (Vovk, 2012). Hence

$$\Pr_{\text{cal}}\big(\text{cov}(r) \geq 1-\alpha\big) = 1 - \text{Beta}_{r,\, k-r+1}(1-\alpha), \tag{18}$$

which is non-decreasing in r. Define $r^\star(k,\alpha,\delta) = \min\{r \leq k : 1 - \text{Beta}_{r,k-r+1}(1-\alpha) \geq 1-\delta\}$ (and $r^\star = \infty$, i.e. $Q^{\text{lcb}} = +\infty$ and $C(x) = \mathcal{Y}$, if no such r exists). Then $Q^{\text{lcb}} = t_{(r^\star)}$ satisfies $\Pr_{\text{cal}}(\text{cov} \geq 1-\alpha) \geq 1-\delta$.

Now let $\hat{q}^{\text{cons}}_{\text{SASC}} = \hat{q}_{\text{SASC}} \vee Q^{\text{lcb}}$ for *any* statistic $\hat{q}_{\text{SASC}}$, including one computed with a data-dependent shrinkage weight. Because $\hat{q}^{\text{cons}}_{\text{SASC}} \geq Q^{\text{lcb}}$ pointwise and $q \mapsto \{y : s(x,y) \leq q\}$ is monotone in q, we have $C^{\text{cons}}(x) \supseteq C_{r^\star}(x)$ for every x. Hence $F_T(\hat{q}^{\text{cons}}_{\text{SASC}}) \geq F_T(Q^{\text{lcb}})$ pointwise on the calibration draw, and the training-conditional guarantee transfers:

$$\Pr_{\text{cal}}\big(\Pr_{\text{test}}(Y \in C^{\text{cons}}(X)) \geq 1-\alpha\big) \;\geq\; \Pr_{\text{cal}}\big(\text{cov}(r^\star) \geq 1-\alpha\big) \;\geq\; 1-\delta. \tag{19}$$

The argument uses nothing about how $\hat{q}_{\text{SASC}}$ was formed, which is the whole point: the floor makes an arbitrary heuristic safe. ■

What happens as $k \to \infty$, corrected. An earlier version of this paper asserted that both forms converge to the naive target quantile. That is false in general, and the failure is instructive. Write b for the anchor's limiting bias, so $\hat{q}_{\text{anc}} \to q_T + b$. The gate accepts the smallest weight whose held-out coverage reaches $1-\alpha$, and as k grows the held-out estimate concentrates on the true coverage of each candidate. If $b > 0$ the pure anchor's limiting coverage is $F_T(q_T + b) > 1 - \alpha$, so $w^\star = 0$ is accepted with probability tending to one and the threshold converges to $q_T + b$, not q_T. Taking the maximum with Q^{lcb} does not remove a positive bias. Only when $b \leq 0$ does every anchored candidate eventually fail the gate, leaving $w^\star = 1$ and the naive quantile. The correct statement is therefore one-sided: limiting coverage is at least $1-\alpha$, and it is exact only when $b \leq 0$.

This is not a hypothetical. Across the 46 settings the median selected weight is 0 at *every* budget we run, including k=500, and the mean rises only from 0.18 to 0.42; 57% of settings have $b > 0$. That is exactly the regime in which the gate keeps the anchor forever, and it is why SASC's threshold MSE plateaus while naive's keeps falling (§D.3). A coverage gate cannot detect over-coverage, because over-coverage is what it is looking for.

The price: rank inflation. The guarantee costs the gap between $r^\star(k,\alpha,\delta)$ and the ordinary split-conformal rank $\lceil (k+1)(1-\alpha) \rceil$; Table 3 lists both. At $\alpha = \delta = 0.1$ the conservative rank is $49/50$ (98th percentile of the target scores instead of the 92nd), and *no* rank suffices at $k = 20$. The guarantee then forces $Q^{\text{lcb}} = +\infty$ and hence the full label set, which produces the degenerate k=20 entries in §4.1. The inflation decays like $O(\sqrt{\alpha(1-\alpha)/k} \cdot z_{1-\delta})$: by the normal approximation to the Beta,

$$\frac{r^\star}{k} \;\approx\; (1-\alpha) + z_{1-\delta}\sqrt{\frac{\alpha(1-\alpha)}{k}}, \tag{20}$$

so the conservative and point forms agree to $O(k^{-1/2})$, and the certified variant becomes practical from a few hundred labels on. Eq. (20) also explains the empirical width ordering of §Q: halving δ costs only the difference of two normal quantiles, so δ is a mild knob compared with k.

Table 3: Training-conditional rank $r^\star(k, \alpha{=}0.1, \delta)$ used by the conservative form, with $r^\star/k$ in parentheses, against the ordinary split-conformal rank $\lceil (k+1)(1-\alpha) \rceil$; "–" means no rank attains the guarantee, so the threshold is $+\infty$ (full sets). The gap to the split rank shrinks as the label budget grows.

k	δ=0.01	δ=0.05	δ=0.10	δ=0.20	split rank
20	–	–	–	20 (1.000)	19
30	–	30 (1.000)	30 (1.000)	29 (0.967)	28
50	50 (1.000)	49 (0.980)	49 (0.980)	48 (0.960)	46
100	97 (0.970)	96 (0.960)	95 (0.950)	94 (0.940)	91
200	190 (0.950)	188 (0.940)	186 (0.930)	185 (0.925)	181
500	466 (0.932)	462 (0.924)	460 (0.920)	457 (0.914)	451
1000	922 (0.922)	916 (0.916)	913 (0.913)	909 (0.909)	901
5000	4550 (0.910)	4536 (0.907)	4528 (0.906)	4519 (0.904)	4501

D.3 Why the anchor helps: a bias–variance characterization

The main text motivates SASC informally: borrow the shape, correct the location. Here we make the trade-off exact, which gives three things the informal argument does not: the size of the anchor's bias in interpretable terms, the label budget beyond which anchoring stops paying, and the optimal shrinkage weight. How that weight behaves in k explains why a *fixed* schedule, such as the $w_k = k/(k+K_0)$ rule we started from, has the wrong functional form.

Define the two estimators of q_T available from k target labels:

$$\hat{q}_{\text{naive}} = \hat{F}^{-1}_{T,k}(1-\alpha), \qquad \hat{q}_{\text{anc}} = q_S + \big(\hat{m}_{T,k} - m_S\big), \tag{21}$$

with $\hat{m}_{T,k}$ the sample median of the k target scores. We treat the source quantities q_S, m_S as known; estimating them adds $O(1/n_s)$ variance, negligible in our regime ($n_s \geq 1500$ versus $k \leq 500$) and quantified at the end of this subsection.

Definition 1 (Anchor bias)**.** $b := \big(q_S - m_S\big) - \big(q_T - m_T\big)$.

b is the difference between the source and target *median-to-tail spreads*. It is zero when the target score law is a location translate of the source's in the range that matters, and it is the only bias the anchor incurs: $\mathbb{E}[\hat{q}_{\text{anc}}] - q_T \to b$. Note that b does *not* measure the size of the shift, since a large but purely translational shift has $b = 0$; this is why SASC can help even when the shift is severe.

Proposition 4 (Asymptotic risk of the anchored and naive quantile estimators)**.** *Under the regularity conditions above, as $k \to \infty$ (Serfling, 1980, Ch. 2),*

$$\mathrm{Var}(\hat{q}_{\text{naive}}) = \frac{v_n}{k} + o(k^{-1}), \qquad v_n = \frac{\alpha(1-\alpha)}{f_T(q_T)^2}, \tag{22}$$

$$\mathrm{Var}(\hat{q}_{\text{anc}}) = \frac{v_a}{k} + O(n_s^{-1}), \qquad v_a = \frac{1}{4 f_T(m_T)^2}, \tag{23}$$

$$\mathrm{Cov}(\hat{q}_{\text{naive}}, \hat{q}_{\text{anc}}) = \frac{c}{k} + o(k^{-1}), \qquad c = \frac{\alpha}{2 f_T(m_T) f_T(q_T)}. \tag{24}$$

Consequently $\mathrm{MSE}(\hat{q}_{\text{anc}}) = b^2 + v_a/k$ *and* $\mathrm{MSE}(\hat{q}_{\text{naive}}) = v_n/k$*, and the anchor has smaller risk if and only if*

$$k \;<\; k^\star := \frac{\kappa}{b^2}, \qquad \kappa := v_n - v_a = \frac{\alpha(1-\alpha)}{f_T(q_T)^2} - \frac{1}{4 f_T(m_T)^2}. \tag{25}$$

Moreover, for the shrinkage family $\hat{q}(w) = w\hat{q}_{\text{naive}} + (1-w)\hat{q}_{\text{anc}}$ the risk-optimal weight is

$$w^\star = \frac{b^2 + (v_a - c)/k}{b^2 + (v_n + v_a - 2c)/k}, \tag{26}$$

which satisfies $w^\star = (v_a - c)/(v_n + v_a - 2c)$, a constant in k, when $b = 0$, and $w^\star = 1 - \Theta\big(1/(kb^2)\big) \to 1$ when $b \neq 0$.

Proof. The variance and covariance expressions are the standard joint asymptotic normality of sample quantiles: for $p \le p'$, $\sqrt{k}(\hat{\xi}_p - \xi_p, \hat{\xi}_{p'} - \xi_{p'}) \Rightarrow N(0, \Sigma)$ with $\Sigma_{pp} = p(1-p)/f(\xi_p)^2$ and $\Sigma_{pp'} = p(1-p')/(f(\xi_p)f(\xi_{p'}))$. Take $p = 1/2$ and $p' = 1-\alpha$, and note $\hat{q}_{\text{anc}}$ is $\hat{\xi}_{1/2}$ plus the constant $q_S - m_S$, which shifts the mean by b and leaves second moments unchanged. The MSE comparison is immediate. For Eq. (26), write $\text{MSE}(w) = (1-w)^2b^2 + w^2v_n/k + (1-w)^2v_a/k + 2w(1-w)c/k$, differentiate, and solve the resulting linear equation; the coefficient of w is $b^2 + (v_n + v_a - 2c)/k > 0$ by Cauchy–Schwarz ($c^2 \le v_n v_a$), so the stationary point is the minimizer. □

Reading the result. Three consequences shape the method. *(i) The anchor's advantage is a tail-versus-center density ratio.* $\kappa > 0$ iff $f_T(m_T)/f_T(q_T) > 1/(2\sqrt{\alpha(1-\alpha)})$, which at $\alpha = 0.1$ means the density at the median exceeds $1.67\times$ the density at the 90th percentile. Score distributions are strongly right-skewed with a sparse upper tail (most points are easy, few are hard), so this holds in practice; §U measures $f_T(m_T)/f_T(q_T)$ across all settings. The estimator borrows strength where the data are dense and spends the labels where they are not. *(ii) There is a finite budget beyond which anchoring must stop.* $k^\star = \kappa/b^2$ is the crossover: for $k \ll k^\star$ the anchor dominates, for $k \gg k^\star$ it is a liability. This matches two empirical findings: SASC's gains are largest at small k and vanish at $k = 500$, and in the 50k-scale experiment (§G) a strong predictor produces a well-estimated but biased anchor. There $|b|$ is large, so $k^\star$ is small and anchoring should be switched off, as the adaptive rule does. §U estimates b, κ and $k^\star$ per setting and compares predicted and observed crossovers. *(iii) A fixed weight schedule has the wrong shape.* Eq. (26) says $w^\star$ is *constant* in k when $b = 0$ and approaches 1 at rate $1/(kb^2)$ otherwise; in both cases it depends on the unknown b. The rule $w_k = k/(k + K_0)$ we tried first depends on k alone, so it is too aggressive when the anchor is good and too timid when it is bad; §P shows this failure across $K_0 \in \{10, 20, 40, 80\}$. It is tempting to read the cross-validated weight of Algorithm 1 as a plug-in for Eq. (26), and an earlier version of this paper did. That reading is wrong and we correct it here. Eq. (26) minimizes squared error and is interior whenever $b \neq 0$; the gate solves a one-sided coverage constraint and returns the *smallest* weight that passes, which is a corner solution. The two disagree in their limits, as §D.2 shows: for $b > 0$ the gate keeps $w = 0$ as $k \to \infty$ while $w^\star \to 1$. It is not an approximation to $w^\star$ under any regime we measure. Empirically the median selected weight is 0 at every budget, whereas $w^\star$ rises with k. We keep Eq. (26) as the statement of when the *anchor* beats the *target-only quantile* in risk, which is what §D.3 tests, and we make no claim that the deployed rule tracks it.

From threshold error to coverage error. Coverage is $F_T(\hat{q})$, so a first-order expansion around q_T gives

$$\text{cov}(\hat{q}) - (1-\alpha) \;=\; f_T(q_T)\,(\hat{q} - q_T) + O\big((\hat{q} - q_T)^2\big). \tag{27}$$

The tail density converts threshold error into coverage error. Two sanity checks follow. For the naive estimator, the standard deviation of the coverage error is $f_T(q_T)\sqrt{v_n/k} = \sqrt{\alpha(1-\alpha)/k}$: the binomial rate, independent of the score distribution, and equal to 4.2 points at k=50. It is the quantitative form of the reliability problem SASC addresses. For the anchor, the coverage error is $f_T(q_T)b + O_p(k^{-1/2})$: a systematic offset proportional to the anchor bias *scaled by the tail density*, which is small when the tail is sparse. So a mildly mis-specified anchor damages coverage less than the same threshold error would in a dense region, a further reason why the anchored estimator suits conformal calibration in particular, rather than quantile estimation in general.

The neglected source term. Estimating q_S and m_S from n_s source points adds $\text{Var} = \big[\alpha(1-\alpha)/f_S(q_S)^2 + 1/(4f_S(m_S)^2) - \alpha/(f_S(m_S)f_S(q_S))\big]/n_s$ to $\text{Var}(\hat{q}_{\text{anc}})$. This term is $O(1/n_s)$ against the anchor's own $O(1/k)$, so it is negligible when $n_s \gg k$ and leaves the analysis above unchanged. §R confirms empirically that shrinking n_s from 3000 to 250 keeps the ordering intact until n_s approaches k.

Finite-sample accuracy of the variance formulae. Prop. 4 is asymptotic in k. Table 4 compares the empirical variance of the naive quantile over 20 resamples per setting with the formula v_n/k, using target-test densities from a kernel estimate and from a quantile-spacing estimate that does not smooth the tail. The ratio is 0.8–0.9 for $k = 20$–200: the asymptotic formula overstates the variance by 10–20% at the smallest budgets and is otherwise accurate. (At k=500 the resample equals the whole labeled pool, so the resampling variance collapses and the ratio is not informative.)

Table 4: Ratio of the empirical variance of the naive quantile to the asymptotic value v_n/k of Prop. 4, median over 46 settings and 5 seeds. The formula overstates the variance slightly at the smallest budgets and is otherwise accurate.

model	estimator	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	naive, kde	0.82	0.91	0.93	0.89	0.92	0.53
TabICLv2	naive, spacing	0.79	0.89	0.90	0.87	0.89	0.50
TabPFNv2	naive, kde	0.86	0.87	0.93	0.90	0.87	0.54
TabPFNv2	naive, spacing	0.84	0.89	0.94	0.89	0.86	0.59

D.4 How much can the point form undercover?

The point form has no distribution-free guarantee, because selecting w by cross-validation on the same k labels that build the quantile breaks exchangeability. This subsection replaces that blanket statement with two precise ones: a lower bound on realized coverage (Prop. 5), and a calculation of how often the inner check can be fooled.

Lemma 1 (Convexity floor)**.** *For any $w \in [0,1]$, $\hat{q}(w) \geq \min\{\hat{q}_{\text{naive}}, \hat{q}_{\text{anc}}\}$, hence $F_T(\hat{q}(w)) \geq \min\{F_T(\hat{q}_{\text{naive}}), F_T(\hat{q}_{\text{anc}})\}$.*

Proof. A convex combination of two reals is at least their minimum; F_T is non-decreasing. □

Proposition 5 (Anchor-bias-limited coverage)**.** *Suppose $f_T \leq f_{\max}$ and $f_T \geq f_{\min} > 0$ on an interval containing m_T, q_T and the realized thresholds. Then for any $\eta \in (0,1)$, with probability at least $1-\eta$ over the k target calibration scores, the realized coverage of* SASC *with* any *weight $w \in [0,1]$, including the cross-validated one, satisfies*

$$\operatorname{cov}\big(\hat{q}(w)\big) \;\geq\; 1-\alpha \;-\; \max\left\{\varepsilon_k,\;\; f_{\max}|b| + \frac{f_{\max}}{f_{\min}}\varepsilon_k\right\}, \qquad \varepsilon_k = \sqrt{\frac{\log(2/\eta)}{2k}}. \tag{28}$$

Proof. By the DKW inequality with Massart's constant (Massart, 1990), $\sup_x |\hat{F}_{T,k}(x) - F_T(x)| \leq \varepsilon_k$ with probability $\geq 1-\eta$. On that event: (i) for the naive threshold, $F_T(\hat{q}_{\text{naive}}) \geq \hat{F}_{T,k}(\hat{q}_{\text{naive}}) - \varepsilon_k \geq 1-\alpha-\varepsilon_k$; (ii) for the median, $|F_T(\hat{m}_{T,k}) - 1/2| \leq \varepsilon_k$, so by the mean value theorem $|\hat{m}_{T,k} - m_T| \leq \varepsilon_k / f_{\min}$, whence $\hat{q}_{\text{anc}} \geq q_T + b - \varepsilon_k/f_{\min}$ and, using $F_T(q_T+u) \geq 1-\alpha+f_{\min}u$ for $u \geq 0$ and $F_T(q_T+u) \geq 1-\alpha-f_{\max}|u|$ for $u<0$, $F_T(\hat{q}_{\text{anc}}) \geq 1-\alpha-f_{\max}(|b| + \varepsilon_k/f_{\min})$. Apply Lemma 1. □

The bound is informative but *not* distribution-free: it depends on the unknown b and on density bounds. This is the status of the point form, a heuristic with a controlled failure mode, and it is why we supply the conservative variant, whose guarantee (Prop. 3) needs none of these quantities. The bound does explain why SASC is never much worse than naive in our experiments: the second term degrades in $|b|$ gracefully rather than catastrophically, and the CV gate removes the large-$|b|$ cases.

How often can the inner check be fooled? The selection rule accepts a weight w when its F-fold held-out coverage estimate $\widehat{\text{cov}}(w)$ is at least $1-\alpha$. That estimate is an average of k indicator variables (each target point is held out exactly once), so $\text{sd}(\widehat{\text{cov}}) \approx \sqrt{\text{cov}(1-\text{cov})/k}$, ignoring the mild dependence from refitting the threshold per fold. A candidate whose *true* coverage is $1-\alpha-\epsilon$ is thus accepted with probability approximately

$$\Phi\left(-\epsilon \Big/ \sqrt{\tfrac{\alpha(1-\alpha)}{k}}\right), \tag{29}$$

which at $\alpha = 0.1$ equals 24% for a 3-point deficit at $k = 50$, 8% at $k = 100$, and 0.4% at $k = 500$; at $k = 20$ the same deficit is accepted 36% of the time. This accounts for the one regime where we see SASC slip below naive, namely $k = 20$ at 50k source scale (§G). The fix is more inner samples (a larger k, or a smaller F with more data per fold) or else the conservative floor. The failure is bounded in practice because the accepted deficit is $O_p(k^{-1/2})$, the same order as naive's own fluctuation, rather than the $O(1)$ deficit an ungated anchor can incur.

D.5 What the oracle gap does and does not certify

Proposition 6 (The oracle gap bounds the whole family)**.** *Let $\mathcal{T}$ be the temper grid and let $\mathcal{A}$ be any selection rule mapping observed data, including the test labels if one wishes, to a temper $\hat{\tau} \in \mathcal{T}$. Then the realized coverage satisfies* $\mathrm{cov}(\hat{\tau}) \leq \max_{\tau \in \mathcal{T}} \mathrm{cov}(\tau) = \mathrm{cov}(0) + \Delta_{\max}$*, where* $\Delta_{\max} = \max_{\tau} \mathrm{cov}(\tau) - \mathrm{cov}(0)$.

Proof. Immediate: $\hat{\tau} \in \mathcal{T}$ and the maximum over $\mathcal{T}$ dominates any particular element. □

Two different tempers, and which one bounds what. The bound above uses the *maximum-coverage* temper. Elsewhere we report the *oracle temper*, the τ whose coverage is closest to nominal, because that is the temper a perfect selection rule would actually want: overshooting $1 - \alpha$ is not a success. These are different objectives and in general different tempers, so the closest-to-nominal oracle does *not* upper-bound coverage over the grid, and fully weighted conformal ($\tau = 1$) exceeds it on some pairs. We therefore report both: Δ for "how close to nominal can any member get" and $\Delta_{\max}$ for the bound of Prop. 6. Only $\Delta_{\max}$ licenses a statement about the whole family, and only Δ licenses a statement about attainable calibration. Earlier versions of this paper conflated them. Simple as it is, this is the logic of §4.2 and Table 18. Measuring Δ costs one sweep over $\mathcal{T}$ with test labels, and a small Δ then rules out *every* method in the family at once, including ones not yet invented, not just the selection rule we implemented. Combined with Prop. 2, which identifies the un-removable term, it turns "our method did not help" into "nothing of this type can help here."

We tried to identify concept shift directly, and could not. A referee observed, correctly, that the failure of a grid of tempered logistic-regression weights does not by itself identify concept shift: a misspecified density ratio or insufficient overlap would look the same. We therefore attempted three estimator-free tests. All three failed their own negative control, and we report that rather than a result we cannot defend.

The control throughout is the induced-shift family of §4, where the target is resampled on X alone, so $P(Y \mid X)$ is unchanged by construction and any valid test must return nothing.

(i) Does Y help a domain classifier? Under pure covariate shift the domain is independent of Y given X, so appending Y to a domain classifier cannot raise its AUC. On ACS the gain was $+0.009$ ($p = 5 \times 10^{-7}$), but on the control it was *larger*, $+0.035$. The reason is visible in the data: across settings the gain correlates -0.81 with the AUC of the X-only classifier. When that classifier underfits, Y proxies for the X-signal it missed, and the gap opens with no conditional shift at all.

(ii) A conditional randomization test. Comparing the same statistic against a null built by resampling labels in both domains from one shared conditional model should cancel that bias, since the underfit enters the null identically. It does not work here, because the only conditional model available is the source-trained TFM, and that model is itself wrong on the target: under the induced shift its accuracy falls from 0.850 to 0.715 and its ECE rises from 0.031 to 0.100. Resampled labels then differ from real ones in the target domain even when the true conditional is unchanged. The control showed an excess of $+0.028$ against ACS's $+0.026$.

(iii) Cross-domain versus within-domain conditional disagreement. Fit $P(Y \mid X)$ on a source half and on a target half, compare their predictions on held-out target covariates, and calibrate against two models fitted on two disjoint source halves of the same size. On ACS the cross-domain disagreement exceeds the within-domain floor by $+0.024$ (39 of 46 settings, $p = 3 \times 10^{-8}$). But the control cannot be run at comparable power: the induced family's shifted region leaves only five settings with enough target points to fit and evaluate, at about 193 points per fit, and on those the excess is $+0.022$, statistically indistinguishable from the ACS value. Fitting a conditional inside a region the source barely covers measures extrapolation as much as conditional shift, which is the overlap problem the referee raised, now biting the diagnostic itself.

We therefore do not claim to have identified concept shift. What the evidence supports is the narrower statement made in §4.2: the reweighting procedures we evaluated do not recalibrate on these shifts, and no temper of our estimated ratios does. Distinguishing an unreachable conditional from an unreachable density ratio would need a setting with a known ground-truth ratio, or overlap good enough to fit both conditionals reliably, and we have neither.

Three caveats, and how we address them. (i) We compute Δ with *estimated* weights $\hat{w}$; if $\hat{w}$ were uninformative, Δ would be small for a trivial reason. We check this directly: the domain classifier separates source from target well above chance on every ACS pair, and $n_{\text{eff}}^{\text{frac}}$ is far from 1 (§U, §U.1), so the weights do carry covariate information; they just do not carry *coverage* information. (ii) The family is tempered weights w^τ, and a richer covariate-only family might do better in principle. Prop. 2 says any such family would still leave a genuine concept term untouched, but it cannot tell us whether our residual is that term. The Wine setting is suggestive rather than decisive: its covariate marginals are nearly disjoint, so covariate information is maximal and overlap is worst, and the oracle misses nominal by 0.10; both a conditional change and an unestimable ratio would produce that. (iii) Δ is estimated on a finite test set and is thus noisy; we average over seeds and report per-setting values in §J. None of these caveats changes the practical conclusion. We use the diagnostic to decide *whether to spend labels*, and its failure mode is to declare covariate methods useless when they would have helped a little. That is the safe direction, since collecting a few hundred target labels is valid regardless.

D.6 Stronger density-ratio estimators

The main text estimates $w = \mathrm{d}P_{\text{test}}/\mathrm{d}P_{\text{cal}}$ with one balanced logistic domain classifier on standardized features. If the reweighting negative result were an artifact of that estimator, a better ratio would close the gap. We therefore recompute the weights for the same calibration and test rows (the splits are seeded, so they are reproduced exactly) with three stronger estimators: a gradient-boosted domain classifier with two-fold cross-fitting, so that no row is weighted by a classifier that saw it; the same with isotonic probability calibration inside each fold; and TabPFNv2 itself as the cross-fitted domain classifier. Table 5 applies each set of weights to every TFM's cached scores on the 46 settings and reports plain conformal, full weighting, the label-free AUTO-τ selection, the best-calibrated temper and the maximum-coverage temper, on the paper's 11-point temper grid. The stronger estimators separate the domains more sharply (n_{eff} 0.67 for the logistic ratio, 0.44 and 0.37 for the cross-fitted GBDT and TabPFNv2), and the temper closest to nominal improves with them: its tail coverage over settings rises from 0.870 to 0.885–0.889 and it lands within 0.02 of nominal in 76–83% of settings against 69%. But that temper is chosen with test labels. The label-free AUTO-τ selection over-covers by 0.02 to 0.03 at 13% wider sets under every estimator, and full weighting over-covers more. Fifty random target labels give the same tail coverage (0.877) at 10% narrower sets. A sharper ratio buys width, not a deployable calibration.

D.7 Is the residual label shift?

Some real shifts, the ACS states in particular, plausibly move the label marginal as well as the covariates. Weighted conformal under label shift (Podkopaev & Ramdas, 2021) reweights the source calibration scores by $w(y) = \pi_T(y)/\pi_S(y)$, with the candidate label's weight as the test mass, so it is the natural baseline for attributing the residual. We run it on the cached predictions with three versions of π_T: black-box shift estimation (Lipton et al., 2018) from the source confusion matrix and the predicted-label marginal on the unlabeled target test set, which is deployable; the empirical marginal of 50 labeled target points; and the oracle target test marginal, which bounds what any label-shift correction can do. Table 6 reports coverage and set size, alongside plain conformal and the covariate density-ratio weights of the main text, and the product of both weightings, the joint-shift correction. The label marginals do move (median total variation 0.05 on ACS and Wine, 0.11 on the UCI family, up to 0.50 on Wine), but correcting for them changes mean coverage by under 0.01 on both families even with the oracle proportions; per setting the oracle correction improves coverage by more than 0.02 in 8% of the ACS settings and 27% of the UCI ones, and never restores nominal. The residual is not label shift either.

E Datasets, shift settings, and per-setting statistics

E.1 The three benchmark families

OpenML suite (induced shift, 24 datasets). This suite serves the diagnosis of §4, the controlled covariate-shift study of §3.2, and the nonconformity-score ablation of §O, since it is the only family in our study with *multiclass* tasks. The datasets are OpenML (Vanschoren et al., 2014) ids 31, 37,

Table 5: Real ACS shift under four density-ratio estimators, per TFM: coverage at nominal 0.90 and set size, averaged over 46 settings and 5 seeds, with the normalized effective sample size of the calibration weights.

		plain		full weight		AUTO-τ		best-calibrated		max coverage		
base model	ratio estimator	cov	size	cov	size	cov	size	cov	size	cov	size	n_{eff}
TabICLv2	logistic (paper)	0.883	1.37	0.915	1.52	0.922	1.53	0.897	1.40	0.927	1.55	0.67
	GBDT, cross-fit	0.883	1.37	0.933	1.59	0.923	1.54	0.905	1.43	0.943	1.61	0.44
	GBDT, cross-fit + isotonic	0.883	1.37	0.902	1.53	0.916	1.53	0.903	1.42	0.938	1.59	0.63
	TabPFNv2, cross-fit	0.883	1.37	0.936	1.60	0.927	1.55	0.903	1.42	0.943	1.61	0.37
TabPFNv2	logistic (paper)	0.874	1.34	0.916	1.52	0.923	1.53	0.895	1.39	0.928	1.55	0.67
	GBDT, cross-fit	0.874	1.34	0.931	1.59	0.924	1.54	0.906	1.42	0.943	1.61	0.44
	GBDT, cross-fit + isotonic	0.874	1.34	0.902	1.53	0.919	1.53	0.904	1.41	0.938	1.59	0.63
	TabPFNv2, cross-fit	0.874	1.34	0.937	1.60	0.929	1.56	0.903	1.41	0.943	1.62	0.37
TabPFNv3	logistic (paper)	0.871	1.33	0.916	1.52	0.923	1.53	0.895	1.38	0.927	1.54	0.67
	GBDT, cross-fit	0.871	1.33	0.932	1.59	0.924	1.54	0.905	1.42	0.943	1.61	0.44
	GBDT, cross-fit + isotonic	0.871	1.33	0.902	1.53	0.918	1.53	0.903	1.41	0.937	1.59	0.63
	TabPFNv2, cross-fit	0.871	1.33	0.936	1.60	0.927	1.55	0.901	1.40	0.943	1.61	0.37
TabDPT	logistic (paper)	0.872	1.32	0.917	1.52	0.923	1.53	0.897	1.39	0.929	1.55	0.67
	GBDT, cross-fit	0.872	1.32	0.932	1.59	0.925	1.54	0.907	1.43	0.943	1.61	0.44
	GBDT, cross-fit + isotonic	0.872	1.32	0.903	1.53	0.921	1.54	0.905	1.41	0.938	1.59	0.63
	TabPFNv2, cross-fit	0.872	1.32	0.937	1.60	0.929	1.56	0.902	1.41	0.943	1.62	0.37
GAMformer	logistic (paper)	0.899	1.44	0.922	1.56	0.912	1.51	0.901	1.45	0.932	1.58	0.67
	GBDT, cross-fit	0.899	1.44	0.932	1.61	0.913	1.52	0.906	1.46	0.944	1.64	0.44
	GBDT, cross-fit + isotonic	0.899	1.44	0.902	1.55	0.911	1.52	0.905	1.46	0.939	1.62	0.63
	TabPFNv2, cross-fit	0.899	1.44	0.937	1.63	0.917	1.53	0.907	1.47	0.944	1.64	0.37

Table 6: Label-shift weighted conformal on the cached predictions, coverage at nominal 0.90 and mean set size averaged over settings and 5 seeds. "both" multiplies the label-shift and covariate density-ratio weights.

		plain (source only)		covariate DR weights		label shift, BBSE		label shift, 50 labels		label shift, oracle π_T		both, oracle π_T	
family	base model	cov	size	cov	size	cov	size	cov	size	cov	size	cov	size
ACS + Wine (46)	TabICLv2	0.883	1.37	0.871	1.34	0.884	1.38	0.881	1.36	0.882	1.36	0.871	1.34
	TabPFNv2	0.874	1.34	0.868	1.33	0.874	1.35	0.872	1.33	0.874	1.33	0.867	1.32
	TabPFNv3	0.871	1.33	0.863	1.31	0.870	1.34	0.869	1.32	0.870	1.32	0.862	1.31
	TabDPT	0.872	1.32	0.864	1.31	0.874	1.34	0.870	1.32	0.871	1.32	0.863	1.30
	GAMformer	0.899	1.44	0.892	1.43	0.903	1.49	0.898	1.43	0.900	1.44	0.893	1.42
UCI family (11)	TabICLv2	0.664	0.96	0.664	0.94	0.683	1.00	0.676	0.98	0.678	0.99	0.666	0.95
	TabPFNv2	0.667	0.97	0.641	0.92	0.692	1.03	0.681	1.00	0.684	1.00	0.637	0.92
	TabPFNv3	0.667	0.97	0.645	0.92	0.686	1.01	0.680	0.99	0.680	1.00	0.636	0.91
	TabDPT	0.657	0.98	0.661	0.94	0.683	1.04	0.665	0.99	0.666	0.99	0.652	0.93
	GAMformer	0.722	1.26	0.763	1.43	0.732	1.30	0.698	1.24	0.699	1.24	0.725	1.31

54, 1067, 1464, 1489, 15, 23, 29, 44, 50, 188, 458, 469, 1050, 1068, 1462, 1480, 1487, 1494, 1510, 40975, 40984, 40994; Table 7 gives rows, features and class counts. We apply two shift protocols. *(a) Hard split* (diagnosis): we sort by the highest-variance numeric feature, train and calibrate on the low region, and test on the high region, a severe, near-disjoint covariate shift. *(b) Exponential tilt* (AUTO-τ study, and the `ind` family of §O and §T): we draw the test pool without replacement with probability $\propto \exp(\beta z)$ in the standardized highest-variance feature z. Support stays overlapping and β sets the severity. In the `ind` family we further split the shifted region into a labeled target pool (the k budget) and a test set, exactly mirroring the concept-shift protocol so the same calibration methods apply.

ACS / folktables (real shift, 45 settings). We take five person-level prediction tasks from the US Census American Community Survey (Ding et al., 2021): income, employment, public coverage, mobility and travel time. Each has source California 2018 and nine targets: NY, TX, MS, PR, FL, WA, OK and WV (all 2018, spatial shift) and CA-2014 (temporal shift). The targets span shift severity, from a near-null temporal control (CA-2014) to Puerto Rico, which differs from California in language, economy and demography. Table 8 lists each task's features, target definition and population filter as instantiated by the installed folktables version, so the protocol is reproducible without our code.

Wine (concept shift, 1 setting). We use the wine-quality data of Cortez et al. (2009), as OpenML `wine-quality-red` (source) and `wine-quality-white` (target), binarized at quality ≥ 6. Red and white wines share a feature schema but have nearly disjoint covariate distributions and a different quality–chemistry relationship. Wine is thus the cleanest available demonstration that low covariate overlap does not imply covariate reweighting helps.

The last two families give the 46 concept-shift settings (the `cs` family) used for all SASC results.

E.2 Preprocessing

We ordinal-encode categorical columns (unseen categories $\to$ NaN $\to$ column median) for the domain classifier, TabPFN, and the non-foundation controls. TabICL takes raw frames, since it does its own column embedding. For the regression study we $\log(1+\cdot)$-transform continuous ACS income. We subsample source pools to $n_{\text{tr}} + n_{\text{cal}} + 3000$ rows and split them into an in-context support set and a *stratified* calibration set, and the target pool to $k_{\max} + n_{\text{te}}$ rows, split into the labeled budget and the test set. No target label ever enters the support set or the score computation for the test points.

E.3 Per-setting shift statistics

Table 9 reports, for each of the 46 concept-shift settings, TabICLv2's and TabPFNv2's source and target accuracy, the $n_{\text{eff}}^{\text{frac}}$ of the density-ratio weights, the Kolmogorov–Smirnov distance between source and target scores, and the anchor bias b of Definition 1. It is the reference table for §U. Two patterns matter. First, the settings with the largest coverage deficits are not those with the smallest $n_{\text{eff}}^{\text{frac}}$; this is the signature of concept shift. Second, SASC gains most where $|b|$ is small, as Prop. 4 predicts.

Table 7: OpenML dataset cards for the 24-dataset induced-shift suite.

id	name	n	p	#cls	%cat
31	credit-g	1000	20	2	65
37	diabetes	768	8	2	0
54	vehicle	846	18	4	0
1067	kc1	2109	21	2	0
1464	blood-transfusion-serv	748	4	2	0
1489	phoneme	5404	5	2	0
15	breast-w	699	9	2	0
23	cmc	1473	9	3	77
29	credit-approval	690	15	2	60
44	spambase	4601	57	2	0
50	tic-tac-toe	958	9	2	100
188	eucalyptus	736	19	5	26
458	analcatdata_authorshi	841	70	4	0
469	analcatdata_dmft	797	4	6	100
1050	pc3	1563	37	2	0
1068	pc1	1109	21	2	0
1462	banknote-authenticatio	1372	4	2	0
1480	ilpd	583	10	2	10
1487	ozone-level-8hr	2534	72	2	0
1494	qsar-biodeg	1055	41	2	0
1510	wdbc	569	30	2	0
40975	car	1728	6	4	100
40984	segment	2310	18	7	0
40994	climate-model-simulati	540	18	2	0

F Protocol, compute, and reproducibility

The code, the cached base-model predictions and every result CSV are released anonymously at `https://anonymous.4open.science/r/TFM-1-ICLR2027-576E/` with checksums and one-command reproduction scripts.

F.1 Protocol

Induced shift. Highest-variance numeric feature; exponential tilt $\propto \exp(\beta z)$ with overlapping support ($\beta \in \{0, 1, 2, 3.5\}$), or a hard low/high split for the diagnosis. Calibration is stratified by

Table 8: ACS task specifications as instantiated by the installed folktables release: number of predictors, binary target definition, and predictor list. Source is always California 2018; each task also applies its folktables population filter (e.g. age > 16 and positive working hours for income).

task	#feat.	positive class	predictors
income	10	PINCP $> 50{,}000$	AGEP, COW, SCHL, MAR, OCCP, POBP, RELP, WKHP, SEX, RAC1P
employment	16	ESR $= 1$ (employed)	AGEP, SCHL, MAR, RELP, DIS, ESP, CIT, MIG, MIL, ANC, NATIVITY, DEAR, DEYE, DREM, SEX, RAC1P
pubcov	19	PUBCOV $= 1$	AGEP, SCHL, MAR, SEX, DIS, ESP, CIT, MIG, MIL, ANC, NATIVITY, DEAR, DEYE, DREM, PINCP, ESR, ST, FER, RAC1P
mobility	21	MIG $= 1$ (same house)	AGEP, SCHL, MAR, SEX, DIS, ESP, CIT, MIL, ANC, NATIVITY, RELP, DEAR, DEYE, DREM, RAC1P, GCL, COW, ESR, WKHP, JWMNP, PINCP
traveltime	16	JWMNP > 20 min	AGEP, SCHL, MAR, SEX, DIS, ESP, MIG, RELP, RAC1P, PUMA, ST, CIT, OCCP, JWTR, POWPUMA, POVPIP

class. **Density ratio.** Balanced ℓ_2-regularized logistic domain classifier on standardized features (C=1), with the class-prior correction that makes the output an estimate of $\mathrm{d}P_T/\mathrm{d}P_S$; PCA-16 for the embedding-space ablation. **Base models.** Frozen TabICLv2 and TabPFNv2, plus XGBoost, random forest and ℓ_2 logistic regression as non-foundation controls (§T). We subsample support sets to n_{tr}=3000 rows. **Levels.** $\alpha = 0.1$ in the main text; $\alpha \in \{0.05, 0.1, 0.2\}$ in §N. **Budgets.** $k \in \{20, 30, 50, 100, 200, 500\}$. **Seeds.** Three seeds in the main-text tables (as originally run); **five** seeds for every appendix study built on the artifact dumps, so appendix numbers can differ from main-text numbers in the third decimal. Seeds control the source/target subsampling, the support/-calibration split and the base model's internal randomness.

F.2 Compute

All experiments run on a single node with H100 GPUs and 16 CPU cores. Table 10 reports fit and inference times per setting for each base model and the total cost of the appendix suite. First, under the artifact-dump design the whole ablation program (eleven studies, several hundred thousand conformal calibrations) is *pure post-processing* that costs minutes of CPU. We call the base model once per (setting, seed) and cache its predictions; re-running every ablation end-to-end would cost roughly two orders of magnitude more GPU time. Second, SASC is free at deployment: its cost is a sort and a five-fold scan over at most 500 scalars, negligible next to a single TFM forward pass (Table 10, last column).

F.3 Reproducibility checklist

- **Code.** We release all scripts: `dump_scores.py` and `dump_reg.py` (artifact dumps), `ablations.py` (all appendix studies), `csconformal.py` (SASC and baselines), `autotau.py` (AUTO-τ), `natural.py` (ACS), `fewshot.py`, `conceptshift.py` (class-conditional), `regression.py`, `extra_bench.py` (Wine), `sherpa.py` (negative ablations), `shift_pilot.py` (diagnosis), `baselines.py` (temperature scaling), `export_for_r.py`, the table generators (`maketables.py`, `maketables2.py`) and the figure script (`figs.R`).
- **Figures.** One R script (`figs.R`, R 4.4 with `ggplot2`) produces all figures from the same CSVs the tables use. Each figure is rendered at text width, so LaTeX never rescales it, and every panel uses one font size (true sub- and superscripts excepted), so type size is uniform across the paper.
- **Data.** We download ACS once via folktables and cache it locally (1.2 GB), and cache OpenML datasets as pickles because `fetch_openml` is not available from compute nodes. Wine is OpenML `wine-quality-{red,white}`. We use no proprietary data.
- **Models.** We use the TabICLv2 and TabPFNv2 checkpoints frozen, with offline HuggingFace caches: no fine-tuning, no pretraining, and no target data in any support set.
- **Determinism.** Every experiment is seeded, and the inner cross-validation of Algorithm 1 uses a fixed permutation seed, so SASC is a deterministic function of its inputs.

Table 9: Per-setting statistics for the 46 concept-shift settings (mean over 5 seeds). acc_s/acc_t: source/target accuracy; $n_{\text{eff}}^{\text{frac}}$: normalized effective sample size of the density-ratio weights; KS: Kolmogorov–Smirnov distance between source and target LAC score distributions; b: anchor bias (Def. 1).

	TabICLv2		TabPFNv2					
setting	acc_s	acc_t	acc_s	acc_t	$n_{\text{eff}}^{\text{frac}}$	KS	b	$\hat{k}^\star$
employment_spatial_FL	0.816	0.823	0.818	0.824	0.76	0.039	+0.016	23233
employment_spatial_MS	0.816	0.822	0.818	0.821	0.60	0.031	+0.001	4487
employment_spatial_NY	0.816	0.819	0.818	0.819	0.89	0.024	-0.005	5845
employment_spatial_OK	0.816	0.824	0.818	0.823	0.64	0.038	+0.000	4007
employment_spatial_PR	0.816	0.759	0.818	0.761	0.01	0.092	-0.065	508
employment_spatial_TX	0.816	0.820	0.818	0.824	0.80	0.030	-0.010	4324
employment_spatial_WA	0.816	0.828	0.818	0.826	0.80	0.027	+0.009	54885
employment_spatial_WV	0.816	0.817	0.818	0.819	0.44	0.027	+0.007	3725
employment_temporal_CA14	0.816	0.804	0.818	0.806	0.96	0.026	-0.027	1791
income_spatial_FL	0.816	0.769	0.812	0.766	0.77	0.076	-0.011	933
income_spatial_MS	0.816	0.745	0.812	0.741	0.57	0.102	-0.013	$> 10^5$
income_spatial_NY	0.816	0.787	0.812	0.780	0.87	0.057	-0.004	27792
income_spatial_OK	0.816	0.753	0.812	0.753	0.70	0.101	+0.003	24048
income_spatial_PR	0.816	0.705	0.812	0.696	0.84	0.132	-0.041	-75
income_spatial_TX	0.816	0.790	0.812	0.789	0.79	0.054	-0.006	2860
income_spatial_WA	0.816	0.807	0.812	0.800	0.88	0.049	+0.013	2869
income_spatial_WV	0.816	0.749	0.812	0.748	0.60	0.106	-0.002	$> 10^5$
income_temporal_CA14	0.816	0.817	0.812	0.812	0.98	0.022	+0.012	81853
mobility_spatial_FL	0.778	0.739	0.776	0.735	0.73	0.069	-0.006	35200
mobility_spatial_MS	0.778	0.769	0.776	0.758	0.52	0.077	+0.056	216
mobility_spatial_NY	0.778	0.782	0.776	0.777	0.83	0.049	+0.065	91
mobility_spatial_OK	0.778	0.714	0.776	0.708	0.49	0.153	+0.065	26
mobility_spatial_PR	0.778	0.855	0.776	0.854	0.01	0.157	+0.019	3154
mobility_spatial_TX	0.778	0.742	0.776	0.736	0.72	0.068	+0.009	22564
mobility_spatial_WA	0.778	0.697	0.776	0.693	0.82	0.138	+0.039	59
mobility_spatial_WV	0.778	0.765	0.776	0.767	0.39	0.103	+0.100	39
mobility_temporal_CA14	0.778	0.771	0.776	0.769	0.92	0.025	-0.025	1064
pubcov_spatial_FL	0.701	0.739	0.712	0.782	0.93	0.052	+0.009	1644
pubcov_spatial_MS	0.701	0.716	0.712	0.760	0.80	0.050	+0.016	242
pubcov_spatial_NY	0.701	0.711	0.712	0.711	0.97	0.051	-0.023	109
pubcov_spatial_OK	0.701	0.718	0.712	0.757	0.88	0.035	-0.002	3180
pubcov_spatial_PR	0.701	0.651	0.712	0.652	0.82	0.130	+0.081	20
pubcov_spatial_TX	0.701	0.740	0.712	0.788	0.95	0.057	+0.026	$> 10^5$
pubcov_spatial_WA	0.701	0.753	0.712	0.764	0.95	0.090	-0.033	211
pubcov_spatial_WV	0.701	0.729	0.712	0.740	0.72	0.073	-0.041	79
pubcov_temporal_CA14	0.701	0.716	0.712	0.746	0.96	0.035	+0.033	124
traveltime_spatial_FL	0.645	0.609	0.667	0.609	0.90	0.129	-0.051	-2
traveltime_spatial_MS	0.645	0.609	0.667	0.630	0.26	0.176	-0.003	5
traveltime_spatial_NY	0.645	0.704	0.667	0.707	0.60	0.182	-0.097	3
traveltime_spatial_OK	0.645	0.605	0.667	0.632	0.22	0.177	+0.052	375
traveltime_spatial_PR	0.645	0.536	0.667	0.521	0.24	0.250	+0.060	-23
traveltime_spatial_TX	0.645	0.583	0.667	0.582	0.77	0.189	-0.043	-42
traveltime_spatial_WA	0.645	0.620	0.667	0.617	0.19	0.132	+0.006	11
traveltime_spatial_WV	0.645	0.625	0.667	0.647	0.27	0.148	+0.019	61
traveltime_temporal_CA14	0.645	0.643	0.667	0.669	0.95	0.025	+0.003	50702
wine_r2w	0.794	0.493	0.770	0.541	0.00	0.491	+0.165	-18

Table 10: Measured compute. Fit and inference times are per (setting, seed) over the full artifact-dump suite; SASC overhead is the calibration cost added on top of naive few-shot conformal.

base model	device	settings	fit (s)	inference (s)	SASC calib. (ms)
TabICLv2	H100	724	0.38	0.42	< 1
TabPFNv2	H100	715	0.25	0.63	< 1
TabPFNv3	H100	700	0.52	1.78	< 1
TabDPT	H100	700	0.47	0.08	< 1
GAMformer	H100	655	0.43	0.00	< 1
XGBoost	CPU	429	0.43	0.01	< 1
Logistic reg.	CPU	420	0.01	0.00	< 1
total artifact dumps: 4343 (model, setting, seed) triples					

- **Metrics.** Coverage, coverage error, tail (10th-percentile) coverage, validity rate, set size / interval width, and worst-class coverage, all defined in §A. We report the threshold-free metrics as primary, because the validity rate depends on an arbitrary 0.88 cut.

- **Statistical treatment.** Five seeds per setting; bootstrap confidence intervals and paired Wilcoxon tests across settings in §X.

F.4 Reproduction commands

In this section, we provide the exact commands that regenerate the paper from the released repository, at the two levels of §F: every table and figure from the shipped result files, and every study from the cached base-model predictions. Figure 6 lists them as run.

```
# Level 1: every table and figure from the shipped result CSVs (CPU only, ~3 min)
python3.12 -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt
tar xzf artifacts/results.tar.gz
python  figures_tables/make_tables.py        # -> figures_tables/tables/*.tex (88 tables)
python  figures_tables/make_tables_pilot.py  # dataset cards and the pilot-phase tables
Rscript figures_tables/make_figures.R        # -> figures_tables/figures/*.pdf
Rscript figures_tables/make_fig_story.R      # -> fig_story.pdf
tectonic figures_tables/fig_pipeline.tex     # -> fig_pipeline.pdf (any XeLaTeX works)

# Level 2: every study from the cached predictions (CPU only, ~3 h on 16 cores)
for t in artifacts/dumps/*.tar.gz; do tar xzf "$t"; done   # -> dumps/{cs,ind,ext,ms,reg,ratios}
export DUMP_ROOT=dumps OMP_NUM_THREADS=1
python src/ablations.py --family cs  --studies main scoreanatomy anatomy theory stats \
                                               alpha score hyper ncal delta cc weights
python src/ablations.py --family ext --studies main scoreanatomy stats cc anatomy
python src/ablations.py --family ind --studies main scoreanatomy anatomy score
python src/ablations.py --family cs  --studies matched   # width-matched control, 10 draws
python src/ablations.py --family ext --studies matched
python src/improve.py   --family cs  --studies cond active ltt semi theory msrc matchedcond
python src/improve2.py  --family cs  --studies finite klabels lttsplit block active1 \
                                               condsens wccfull concept crt condcmp
python src/labelshift.py cs && python src/labelshift.py ext   # label-shift weighted conformal
python src/ratio_eval.py                    # four density-ratio estimators (weights shipped)
BOOT_B=2000 BLOCK=setting python src/matched_boot.py marginal     # paired cluster bootstrap
BOOT_B=2000 BLOCK=task    python src/matched_boot.py marginal
BOOT_B=2000 BLOCK=setting python src/matched_boot.py conditional
BOOT_B=2000 BLOCK=task    python src/matched_boot.py conditional
for M in nexcp pooled_dr wcc_cspd; do                              # the other rules
  METHOD=$M BOOT_B=1000 KS="20,50,100" TAILS="per draw" BY=size \
    BLOCK=setting python src/matched_boot.py marginal
  METHOD=$M BOOT_B=1000 KS="20,50,100" TAILS="per draw" BY=size \
    BLOCK=task    python src/matched_boot.py marginal
done

# Level 3: the cached predictions themselves (one GPU per job)
HARNESS_MODEL=tabicl HARNESS_DEV=cuda python src/dump_scores.py --family cs --seeds 0 1 2 3 4
#   HARNESS_MODEL in {tabicl, tabpfn, tabpfn3, tabdpt, gamformer, xgb, logreg}
#   --family in {cs, ind, ext}
```

Figure 6: Reproduction commands, as run, for the three levels of the released repository.

F.5 Calibration rules and the control, as configured

In this section, we provide the configuration of every calibration rule, of the width-matched control and of its bootstrap, exactly as the released code sets them (Figure 7). Every value is a constant in `ablations.py`, `autotau.py` or `matched_boot.py`.

F.6 Shift settings, as instantiated

In this section, we provide the settings exactly as the dump scripts instantiate them: the 46 census and wine settings, the 11 UCI settings, and the sample sizes shared by all of them (Figure 8).

F.7 Density-ratio and label-shift estimators, as run

In this section, we provide the estimators behind §D.6 and §D.7 exactly as run (Figure 9); all of them see the same calibration and test rows, which the seeded splits reproduce.

```
level            alpha = 0.10; conservative form: delta = 0.10
budgets          k in {20, 30, 50, 100, 200, 500}, drawn from a 500-point labeled target pool
score            LAC s = 1 - p(y|x) (classification); |y - yhat| (regression);
                 APS and RAPS in the score sweep

naive few-shot   q = Q_{1-alpha}(t_1..t_k), rank ceil((k+1)(1-alpha))
source anchor    q_src = Q_{1-alpha}(s) + med(t) - med(s)         # median gap, not mean gap
SASC (probe)     q(w) = w * Q_{1-alpha}(t) + (1-w) * q_src
                 w in {0, .10, .25, .40, .55, .70, .85, 1}, scanned ascending (most anchor first)
                 gate: 5-fold held-out target coverage >= 1 - alpha; fixed fold permutation
                 fallback: w = 1 (naive) if no candidate passes, or if k < 10
SASC-cons        q <- max(q, t_(r)),  r = min{r : 1 - Beta_{r,k-r+1}(1-alpha) >= 1-delta}
conditional SASC tertile bins of max_y p(y|x) on the source calibration set;
                 marginal rule below 8 labels
pooled           plain quantile of {s + (med(t) - med(s))} U {t}
pooled-DR        weighted quantile of {s} U {t}; source weights w(x)/mean(w), target weight 1
WCC-CSPD         per class j: class-j source scores at weight w(x) * P_S(j)/P_T(j)
                 pooled with class-j target scores;  y in C(x) iff 1 - p_j(x) <= t_j
NexCP-gated      one weight rho on every source point, rho in {0, .05, .10, .25, .50, 1},
                 chosen by the same 5-fold gate
fixed schedule   w_k = k / (k + K0), K0 = 40 (appendix only)

AUTO-tau         w -> w^tau, tau in {0, .05, .10, .15, .20, .30, .40, .55, .70, .85, 1};
                 split the source calibration set in half; importance-weighted coverage
                 on the held-out half; smallest tau whose lower confidence bound
                 (z = 1, n_eff-calibrated) reaches 1 - alpha
oracle temper    tau closest to nominal coverage on the target test set;
                 the maximum-coverage temper is reported separately

control          comparator: naive at alpha' in {.10, .09, .08, .07, .06, .05, .04, .03, .02}
                 matched on mean set size (or mean coverage) by linear interpolation
                 along the comparator frontier
                 10 independent draws of the k labels per (setting, seed, k);
                 tail = 10th percentile over draws (per draw) or over settings (per setting)
bootstrap        paired cluster bootstrap; frontier and match recomputed in every replicate
                 blocks = settings, or whole ACS tasks with Wine as its own block
                 B = 2000 replicates for SASC; B = 1000 for NexCP-gated, pooled-DR, WCC-CSPD
                 95% percentile intervals
```

Figure 7: Every calibration rule, the width-matched control and its bootstrap, as configured in the released code.

```
sample sizes     support set n_tr = 3000 source rows
                 calibration set n_cal = 3000 source rows, capped at 40% of the source pool
                 (so Wine-red uses 639)
                 labeled target pool k_pool = 500; test set n_te = 2500, disjoint from the pool
                 seeds 0-4 control the subsampling, the support/calibration split and the model

census (45)      source: California 2018, always
                 tasks:   income, employment, pubcov, mobility, traveltime   (folktables ACS)
                 targets: temporal_CA14 (California 2014); spatial_NY, spatial_TX, spatial_MS,
                          spatial_PR, spatial_FL, spatial_WA, spatial_OK, spatial_WV (2018)
                 setting name = <task>_<target>, e.g. income_spatial_PR
wine (1)         wine_r2w: red as source, white as target; label = quality >= 6
UCI family (11)  six public tables, split feature removed from the predictors:
                 adult_nonUS, bank_offpeak, covertype_area2, covertype_area3, covertype_area4,
                 diabetes_emergency, diabetes_transfer, electricity_late, electricity_mid,
                 gasdrift_b4to6, gasdrift_b7to10
induced (24)     OpenML datasets split on the highest-variance feature
                 (low region: train and calibrate; high region: test)
                 controlled covariate shift instead draws the test pool with probability
                 proportional to exp(beta z), beta in {0, 1, 2, 3.5}
```

Figure 8: The shift settings and sample sizes, as instantiated by the dump scripts.

```
density ratio    w(x) = p(x) / (1 - p(x)) * n_cal / n_test,  p(x) = P(target | x)
  logistic       StandardScaler on all rows; LogisticRegression(C = 1, max_iter = 1000),
                 fit in sample (the estimator used in the main text)
  GBDT, cross-fit HistGradientBoostingClassifier(max_iter = 300, learning_rate = 0.05,
                 max_leaf_nodes = 31); 2-fold StratifiedKFold(shuffle, random_state = seed);
                 every row is weighted by the fold that did not see it
  GBDT, isotonic the same inside CalibratedClassifierCV(method = "isotonic", cv = 3),
                 2-fold cross-fit
  TabPFNv2       TabPFNClassifier(random_state = seed) as the domain classifier,
                 2-fold cross-fit, on GPU
  n_eff          (sum w_i)^2 / (n sum w_i^2) over the calibration weights

label shift      w(y) = pi_T(y) / pi_S(y) on the source calibration scores;
                 test mass w(y) for candidate label y
  BBSE           pi_T from the source confusion matrix and the predicted-label
                 marginal on the unlabeled target test set
  pool           pi_T = empirical label marginal of the first 50 labeled target points
  oracle         pi_T = true label marginal of the target test set (an upper bound)
  both           label-shift weights multiplied by the covariate density-ratio weights
```

Figure 9: Density-ratio and label-shift estimators, as run on the cached predictions.

G SCALE EXPERIMENT

We repeat the label-efficiency comparison with a 50k-row source calibration set (TabICL on one H100) on the concept-shift subset (validity rate, 3 seeds):

k	20	50	100	200	500
naive	0.83	0.78	0.94	0.89	0.94
SASC (fixed w)	0.44	0.72	0.89	0.89	0.94
SASC (adaptive)	0.50	0.78	0.94	0.94	0.94

At this scale the 50k-trained predictor is strong, naive few-shot is already reliable at small k, and concept shift biases the high-confidence source anchor. The *fixed-weight* SASC falls below naive. The *adaptive* weight (main text) detects the anchor's failure with its held-out check and falls back to naive, matching it for $k \geq 30$; this rule lets us use SASC without knowing the shift regime in advance. The only residual gap is at k=20, where the 5-fold check over 20 points is itself too noisy. A conservative floor (Prop. 3) or a larger inner sample would fix this.

H PER-DATASET DIAGNOSIS (BOTH MODELS)

Table 11: **Diagnosis.** Accuracy, expected calibration error and split-conformal coverage at nominal 0.90 under an IID split and the induced covariate-shift split, mean over 24 OpenML datasets and 5 seeds. Both models lose 12 to 20 coverage points under shift.

	Accuracy		ECE		Coverage@90	
	IID	shift	IID	shift	IID	shift
TabICLv2	0.850	0.715	0.031	0.100	0.903	0.697
TabPFNv2	0.844	0.728	0.033	0.104	0.902	0.775
TabPFNv3	0.848	0.716	0.029	0.131	0.903	0.737
TabDPT	0.848	0.715	0.030	0.102	0.902	0.733
GAMformer	0.820	0.695	0.045	0.146	0.902	0.775

Figure 10 and the tables below give Table 11 per dataset: accuracy, ECE and split-conformal coverage under the IID and hard-split protocols for the 24 OpenML datasets. The loss of coverage is broad, not confined to a few pathological datasets, and the datasets where coverage collapses furthest are generally not those where accuracy falls furthest. This is the first sign that coverage failure is distributional rather than predictive.

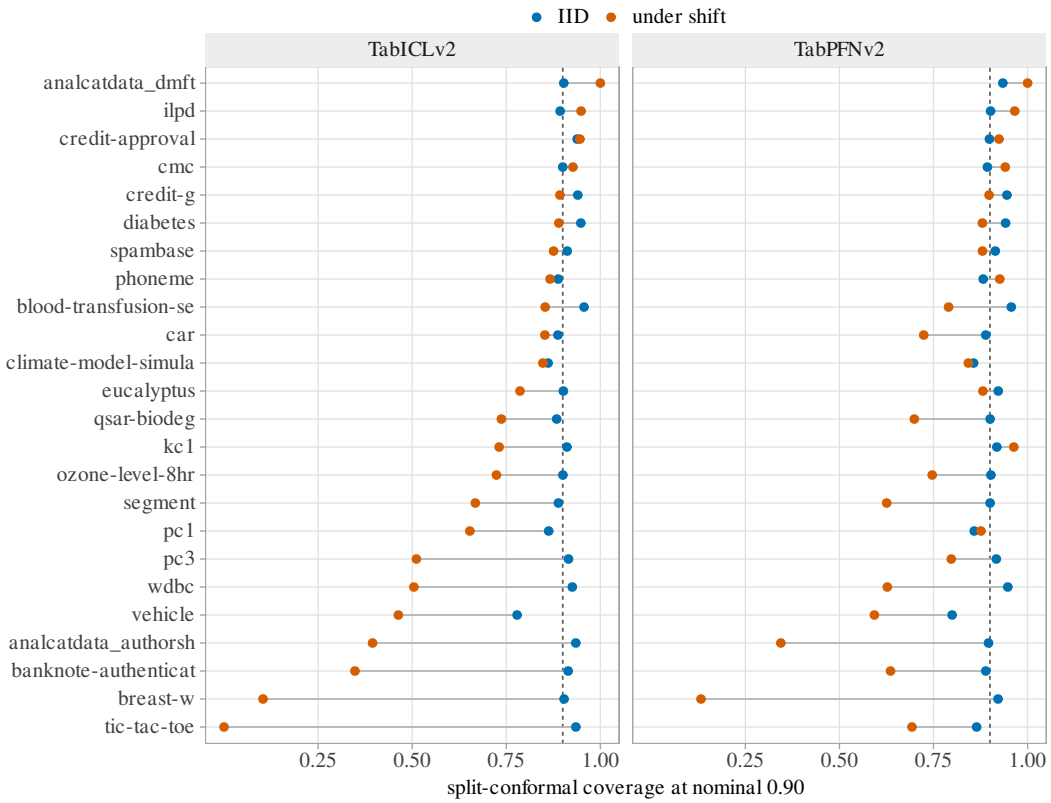


Figure 10: Split-conformal coverage of the 24 OpenML datasets under the IID split (blue) and the covariate-shift split (orange), ordered by the shifted coverage of TabICLv2. Under shift the whole distribution moves left rather than a few outliers.

Table 12: TabICLv2.

id	name	acc		ECE		cov@90	
		iid	sh	iid	sh	iid	sh
31	credit-g	0.75	0.66	0.06	0.08	0.94	0.89
37	diabetes	0.77	0.73	0.06	0.05	0.95	0.89
1464	blood-transfusio	0.85	0.67	0.08	0.12	0.96	0.85
1489	phoneme	0.89	0.72	0.02	0.08	0.89	0.87
54	vehicle	0.81	0.27	0.07	0.43	0.78	0.46
1067	kc1	0.86	0.68	0.04	0.12	0.91	0.73
15	breast-w	0.97	0.84	0.03	0.06	0.90	0.10
29	credit-approval	0.87	0.91	0.03	0.07	0.94	0.95
50	tic-tac-toe	0.99	0.44	0.01	0.27	0.93	0.00
188	eucalyptus	0.64	0.61	0.06	0.13	0.90	0.79
458	analcatdata_aut	1.00	0.95	0.00	0.03	0.93	0.39
469	analcatdata_dmf	0.20	0.20	0.08	0.05	0.90	1.00
1050	pc3	0.90	0.81	0.03	0.04	0.92	0.51
1068	pc1	0.94	0.88	0.02	0.05	0.86	0.65
1462	banknote-authent	1.00	0.83	0.00	0.10	0.91	0.35
1480	ilpd	0.75	0.86	0.08	0.11	0.89	0.95
1494	qsar-biodeg	0.86	0.73	0.07	0.08	0.88	0.74
1510	wdbc	0.96	0.73	0.03	0.06	0.93	0.50
40994	climate-model-si	0.95	0.93	0.02	0.04	0.86	0.85
40975	car	0.98	0.81	0.01	0.07	0.89	0.85
23	cmc	0.56	0.35	0.04	0.16	0.90	0.93
44	spambase	0.95	0.94	0.02	0.13	0.91	0.88
1487	ozone-level-8hr	0.95	0.75	0.01	0.09	0.90	0.72
40984	segment	0.98	0.89	0.01	0.05	0.89	0.67

I AUTO-τ DETAILS AND CONTROLLED-SHIFT RESULTS

AUTO-τ tempers density-ratio weights and selects the temper without target labels (Algorithm 2); its selection criterion is unbiased under covariate shift (Prop. 7). *Why temper:* exact weighted confor-

Table 13: TabPFNv2.

id	name	acc		ECE		cov@90	
		iid	sh	iid	sh	iid	sh
31	credit-g	0.77	0.67	0.06	0.08	0.94	0.90
37	diabetes	0.78	0.75	0.05	0.06	0.94	0.88
1464	blood-transfusio	0.84	0.67	0.08	0.12	0.96	0.79
1489	phoneme	0.87	0.67	0.02	0.14	0.88	0.93
54	vehicle	0.78	0.35	0.06	0.23	0.80	0.59
1067	kc1	0.86	0.70	0.04	0.02	0.92	0.96
15	breast-w	0.97	0.88	0.02	0.06	0.92	0.13
29	credit-approval	0.87	0.90	0.04	0.06	0.90	0.92
50	tic-tac-toe	0.99	0.55	0.09	0.29	0.86	0.69
188	eucalyptus	0.67	0.63	0.04	0.10	0.92	0.88
458	analcatdata_aut	1.00	0.88	0.00	0.03	0.90	0.34
469	analcatdata_dmf	0.18	0.20	0.07	0.02	0.93	1.00
1050	pc3	0.91	0.81	0.02	0.12	0.92	0.80
1068	pc1	0.94	0.88	0.03	0.06	0.86	0.88
1462	banknote-authent	1.00	0.94	0.00	0.16	0.89	0.64
1480	ilpd	0.72	0.87	0.05	0.14	0.90	0.97
1494	qsar-biodeg	0.86	0.68	0.04	0.15	0.90	0.70
1510	wdbc	0.96	0.84	0.02	0.10	0.95	0.63
40994	climate-model-si	0.94	0.94	0.02	0.03	0.86	0.84
40975	car	0.98	0.75	0.03	0.07	0.89	0.72
23	cmc	0.57	0.37	0.02	0.17	0.89	0.94
44	spambase	0.95	0.92	0.02	0.10	0.91	0.88
1487	ozone-level-8hr	0.95	0.78	0.01	0.05	0.90	0.75
40984	segment	0.98	0.91	0.01	0.05	0.90	0.63

mal (τ=1) is unbiased but has high variance: as covariates separate, weights concentrate and n_{eff} collapses, so it over-covers with wide sets. Plain (τ=0) is biased and under-covers. The right τ tracks n_{eff}, and the z-scaled floor turns the noisy point estimate into a high-probability one.

Algorithm 2 AUTO-τ: label-free tempering selection

1: **Input:** calib scores, target covariates, ratio w, grid $\mathcal{T}$, level α, multiplier z
2: split calibration into fit / selection; $n_{\text{eff}} \leftarrow (\sum_j w_j)^2 / \sum_j w_j^2$ on selection
3: **for** $\tau \in \mathcal{T}$ ascending **do**
4: build w^τ-weighted sets on selection using fit-fold scores; $\hat{\mu} \leftarrow \sum_j w_j \mathbf{1}[Y_j \in C(X_j)] / \sum_j w_j$
5: $\widehat{\text{se}} \leftarrow \sqrt{\hat{\mu}(1-\hat{\mu})/n_{\text{eff}}}$; **if** $\hat{\mu} - z\,\widehat{\text{se}} \geq 1-\alpha$ **return** τ
6: **end for**
7: **return** $\max \mathcal{T}$

Proposition 7 (Unbiased selection criterion)**.** *Under covariate shift with shared $P(Y \mid X)$ and ratio w, the self-normalized importance-weighted coverage on calibration-distributed points is consistent for the target coverage $\Pr_{\text{test}}(Y \in C(X))$; selecting the smallest τ with $\hat{\mu} \geq 1-\alpha$ targets nominal coverage as $n_{\text{eff}} \to \infty$. Under concept shift the identity fails and no τ recovers coverage. (Proof: Appendix D.1.)*

Table 14: Controlled covariate shift (24 datasets, 3 seeds). Coverage@90 (AUTO-τ set size in parentheses); both models behave the same.

model	β (n_{eff})	plain	naive	AUTO-τ	oracle
TabICLv2	1.0 (.46)	0.892	0.912	0.906 (1.36)	0.906
	2.0 (.35)	0.855	0.935	0.919 (1.52)	0.905
	3.5 (.31)	0.800	0.953	0.913 (1.72)	0.913
TabPFNv2	2.0 (.35)	0.858	0.935	0.914 (1.54)	0.906
	3.5 (.31)	0.817	0.954	0.913 (1.73)	0.913

Per-dataset (β=2):

J FULL ACS RESULTS, PER TASK/PAIR (BOTH MODELS)

Figure 12 and the following tables give the per-task, per-target-pair values behind Table 17: $n_{\text{eff}}^{\text{frac}}$, the selected temper $\tau^\star$, and the coverage of plain, AUTO-τ, oracle and naive weighted conformal. They

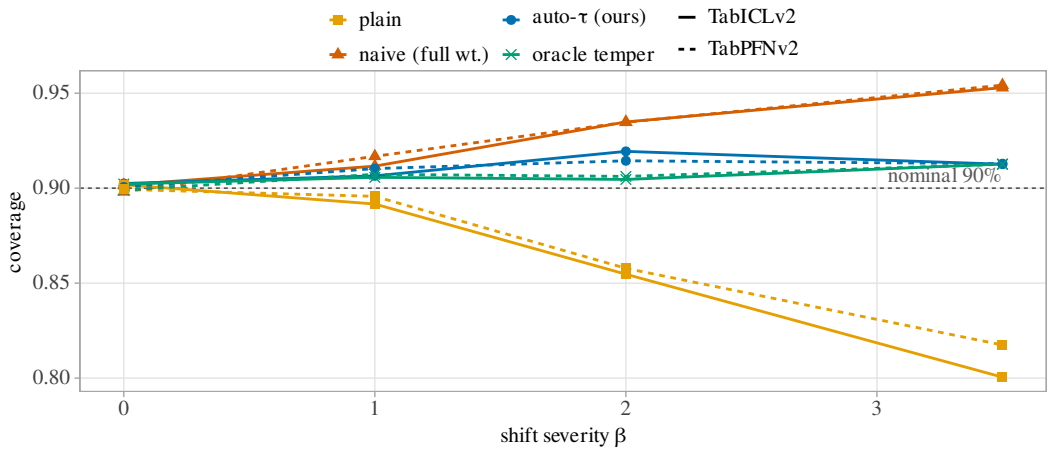


Figure 11: **AUTO-τ under controlled covariate shift.** Coverage against shift severity β on 24 OpenML datasets for plain conformal (yellow), fully weighted conformal (orange), AUTO-τ (blue) and the oracle temper (green); solid lines TabICLv2, dashed TabPFNv2; dashed gray line nominal 0.90. AUTO-τ tracks the oracle without labels when the shift is covariate.

Table 15: TabICLv2.

id	name	plain	AUTO-τ	oracle	AUTO-τ size
15	breast-w	0.72	0.97	0.90	1.66
23	cmc	0.90	0.94	0.91	2.55
29	credit-approval	0.93	0.93	0.92	1.10
31	credit-g	0.89	0.95	0.91	1.67
37	diabetes	0.91	0.93	0.92	1.59
44	spambase	0.90	0.92	0.91	1.10
50	tic-tac-toe	0.70	0.92	0.88	0.94
54	vehicle	0.83	0.89	0.90	1.85
188	eucalyptus	0.93	0.93	0.93	1.62
458	analcatdata_aut	0.83	0.90	0.90	1.33
469	analcatdata_dmf	0.92	0.94	0.92	5.36
1050	pc3	0.89	0.91	0.90	1.06
1067	kc1	0.84	0.92	0.91	1.31
1068	pc1	0.79	0.91	0.90	1.02
1462	banknote-authent	0.79	0.92	0.89	0.97
1464	blood-transfusio	0.91	0.92	0.90	1.57
1480	ilpd	0.91	0.90	0.90	1.37
1487	ozone-level-8hr	0.76	0.90	0.89	0.98
1489	phoneme	0.74	0.84	0.90	1.17
1494	qsar-biodeg	0.90	0.94	0.91	1.57
1510	wdbc	0.93	0.97	0.93	1.40
40975	car	0.81	0.88	0.89	0.96
40984	segment	0.88	0.88	0.90	0.89
40994	climate-model-si	0.90	0.96	0.91	1.51

are the evidence for the decoupling claim of §4.2: reading down the $n_{\text{eff}}^{\text{frac}}$ and coverage columns, the targets with the most covariate movement are not those with the worst coverage.

K Full few-shot results, per task (both models)

The tables below give the per-task detail behind Table 21: how much coverage k labeled target points recover. The gain tracks shift severity: it is largest on the pairs where source-only calibration fails worst and slightly negative on the near-null temporal control, where the extra labels only add variance. This observation motivates gating few-shot calibration on a shift signal (§4.2).

L The appendix evaluation suite

We compute all studies in §M–§X from one set of cached base-model artifacts. For every (base model, setting, seed) triple we store the predicted probabilities on the source calibration set, on the labeled target pool, and on the target test set, together with the density-ratio weights. Every conformal

Table 16: TabPFNv2.

id	name	plain	AUTO-τ	oracle	AUTO-τ size
15	breast-w	0.74	0.95	0.90	1.61
23	cmc	0.90	0.94	0.90	2.50
29	credit-approval	0.93	0.93	0.93	1.12
31	credit-g	0.89	0.95	0.92	1.65
37	diabetes	0.92	0.92	0.92	1.54
44	spambase	0.91	0.92	0.91	1.10
50	tic-tac-toe	0.81	0.85	0.89	0.86
54	vehicle	0.77	0.85	0.88	1.85
188	eucalyptus	0.93	0.93	0.93	1.68
458	analcatdata_aut	0.80	0.90	0.89	1.65
469	analcatdata_dmf	0.94	0.96	0.94	5.63
1050	pc3	0.89	0.90	0.90	1.05
1067	kc1	0.86	0.91	0.90	1.31
1068	pc1	0.81	0.90	0.89	1.01
1462	banknote-authent	0.75	0.95	0.91	1.01
1464	blood-transfusio	0.91	0.92	0.91	1.60
1480	ilpd	0.90	0.90	0.90	1.37
1487	ozone-level-8hr	0.76	0.89	0.90	0.96
1489	phoneme	0.78	0.85	0.90	1.20
1494	qsar-biodeg	0.89	0.92	0.90	1.41
1510	wdbc	0.94	0.96	0.94	1.28
40975	car	0.78	0.88	0.89	0.98
40984	segment	0.88	0.90	0.90	1.05
40994	climate-model-si	0.89	0.96	0.90	1.50

Table 17: **Real ACS shift.** Coverage at nominal 0.90, CA-2018 $\rightarrow$ target, five tasks, nine targets, 5 seeds. Two tempers are reported because they answer different questions: the one closest to nominal is the best *calibration* the family can reach and gains at most 0.026 over plain conformal, while the maximum-coverage one is the bound of Prop. 6 and reaches 0.93 only by over-covering at a 14% width cost.

base model	plain cov / size	AUTO-τ cov	closest-to-nominal cov / size	max-coverage cov / size	fully weighted cov / size
TabICLv2	0.882 / 1.34	0.920	0.899 / 1.38	0.931 / 1.53	0.927 / 1.52
TabPFNv2	0.871 / 1.31	0.918	0.897 / 1.37	0.931 / 1.53	0.927 / 1.52
TabPFNv3	0.872 / 1.31	0.917	0.898 / 1.37	0.931 / 1.53	0.927 / 1.52
TabDPT	0.877 / 1.32	0.919	0.899 / 1.37	0.930 / 1.53	0.926 / 1.52
GAMformer	0.896 / 1.42	0.921	0.900 / 1.43	0.931 / 1.56	0.928 / 1.55

Table 18: **Reliability across shift types** (TabICLv2; TabPFNv2 within 0.01). Δ is the gain of the best-*calibrated* temper, so it measures attainable calibration, not an upper bound on coverage; the bound of Prop. 6 uses the maximum-coverage temper, which on ACS gains 0.049 but only by over-covering. Δ is large only under controlled covariate shift, and few-shot target calibration reaches nominal wherever it is small.

setting	shift type	plain	best-calib.	Δ	few-shot
Induced β=2	covariate	0.855	0.905	**0.050**	–
ACS (5 tasks)	real (mixed)	0.882	0.899	0.017	0.898
ACS regression	real (mixed)	0.877	0.883	0.006	0.900
Wine red$\rightarrow$white	concept	0.701	0.799	0.098†	0.890

† the oracle moves but misses nominal by 0.10: covariates are near-disjoint ($n_{\text{eff}} \approx 0$) yet the deficit is conditional.

method is a function of those arrays, so the comparisons below are *exactly paired*. Naive, pooled, SASC and the conservative variants see the same predictions, calibration points and test points. Any difference therefore comes from the calibration rule alone.

Baselines that use the source. Besides naive few-shot and SASC, the suite runs four rules that pool source and target scores. *Pooled*: the plain conformal quantile of the median-shifted source scores together with the target scores. *Pooled-DR*: weighted conformal with the target scores at weight one and the source scores at their density-ratio weight $w(x)$ (normalized to mean one), with the usual $+\infty$ test mass. *WCC-CSPD* (Wang & Qiao, 2025): per class j, the class-j source scores $1-\hat{p}_j$ enter

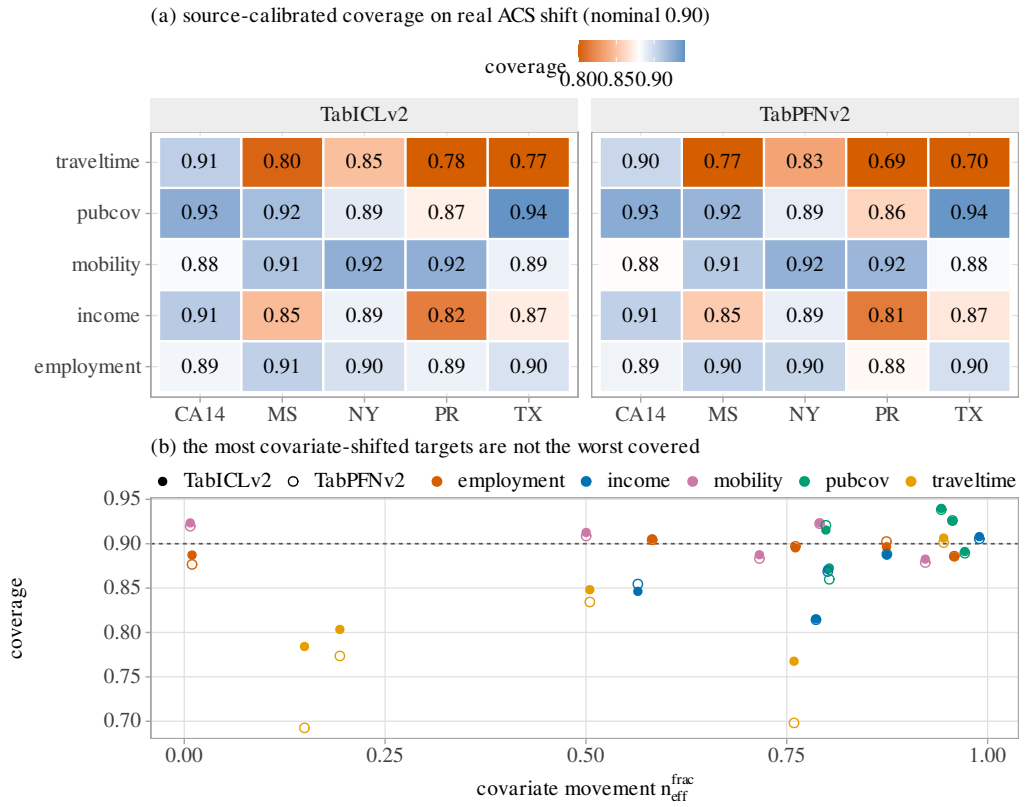


Figure 12: Real ACS shift by task and target. *(a)* Coverage of source-calibrated conformal prediction for each task against each target, both models (blue: at or above nominal; orange: undercoverage). *(b)* The same points against the covariate movement $n_{\text{eff}}^{\text{frac}}$; a covariate-driven deficit would slope down to the left, and it does not.

at weight $w(x)\,P_S(j)/P_T(j)$, which approximates the class-conditional ratio $\mathrm{d}Q_{X|j}/\mathrm{d}P_{X|j}$ under posterior drift, and the class-j target scores enter at weight one. The class-j threshold is the weighted $(1-\alpha)$ quantile, and label j enters the set when $1-\hat{p}_j(x)$ is below it. This follows the original construction on our scores and weights. *NexCP-gated*: non-exchangeable conformal (Barber et al., 2023) with weight ρ on every source score and one on every target score. We scan ρ downward from 1 over $\{1, 0.5, 0.25, 0.1, 0.05, 0\}$ and keep the largest value whose 5-fold held-out target coverage is at least $1-\alpha$, exactly as we choose SASC's weight. NexCP-gated thus differs from SASC only in what it transfers (points rather than a quantile), which is the comparison that separates the gate from the anchor.

The suite shares the protocol of the main text's gated-transfer results: five seeds and a source calibration set of $n_s = 3000$, which §R subsamples. The AUTO-τ and ACS few-shot tables of §I and §K come from the earlier three-seed runs with $n_s = 1500$, so their numbers differ from the suite's in the second or third decimal. The suite also caps calibration at 40% of the source pool. That cap affects only Wine, whose red-wine source has ~ 1.6k rows, so Wine's predictor trains on more rows here than in those earlier tables. The method ordering is unchanged. Where a number differs we report the suite value and say so. Table 26 replicates the main result on the appendix suite; Tables 27 and 28 give mean coverage and set size for the same suite. The wider suite (five seeds, five base models) adds two refinements to the main-text ordering. First, SASC beats naive at every budget for TabICLv2, XGBoost and logistic regression. For TabPFNv2 it beats naive at every budget except k=20, where it is 0.004 lower, a difference the paired test cannot distinguish from zero (§X). Second, the *pooled* baseline is not always the worst option. It hurts the two foundation models, and badly so for TabPFNv2, which it drags to 0.79–0.82. For the two classical base models it slightly *beats* naive at $k \le 50$, because their flatter source score distributions survive pooling better, though SASC is above pooled for all four.

Table 19: TabICLv2.

setting	n_{eff}	plain	AUTO-τ	oracle	naive
employment/spatial_MS	0.58	0.905	0.905	0.905	0.905
employment/spatial_NY	0.87	0.897	0.898	0.898	0.898
employment/spatial_PR	0.01	0.887	0.905	0.897	0.931
employment/spatial_TX	0.76	0.895	0.895	0.896	0.895
employment/temporal_CA14	0.96	0.885	0.885	0.885	0.885
income/spatial_MS	0.56	0.846	0.854	0.855	0.855
income/spatial_NY	0.87	0.887	0.888	0.889	0.889
income/spatial_PR	0.79	0.815	0.817	0.817	0.817
income/spatial_TX	0.80	0.871	0.875	0.878	0.878
income/temporal_CA14	0.99	0.908	0.906	0.906	0.906
mobility/spatial_MS	0.50	0.913	0.904	0.903	0.899
mobility/spatial_NY	0.79	0.924	0.924	0.924	0.926
mobility/spatial_PR	0.01	0.923	0.931	0.915	0.872
mobility/spatial_TX	0.72	0.888	0.891	0.894	0.894
mobility/temporal_CA14	0.92	0.883	0.887	0.887	0.887
pubcov/spatial_MS	0.80	0.915	1.000	0.915	1.000
pubcov/spatial_NY	0.97	0.891	1.000	0.896	1.000
pubcov/spatial_PR	0.80	0.873	0.964	0.894	1.000
pubcov/spatial_TX	0.94	0.940	0.979	0.940	1.000
pubcov/temporal_CA14	0.96	0.925	0.922	0.922	0.923
traveltime/spatial_MS	0.19	0.803	1.000	0.889	1.000
traveltime/spatial_NY	0.50	0.848	1.000	0.886	1.000
traveltime/spatial_PR	0.15	0.784	1.000	0.881	1.000
traveltime/spatial_TX	0.76	0.768	0.960	0.918	1.000
traveltime/temporal_CA14	0.95	0.906	0.908	0.906	0.908

Table 20: TabPFNv2.

setting	n_{eff}	plain	AUTO-τ	oracle	naive
employment/spatial_MS	0.58	0.904	0.900	0.903	0.902
employment/spatial_NY	0.87	0.902	0.903	0.901	0.901
employment/spatial_PR	0.01	0.877	0.912	0.899	0.908
employment/spatial_TX	0.76	0.897	0.896	0.897	0.896
employment/temporal_CA14	0.96	0.886	0.885	0.886	0.885
income/spatial_MS	0.56	0.855	0.859	0.861	0.861
income/spatial_NY	0.87	0.888	0.890	0.891	0.890
income/spatial_PR	0.79	0.814	0.816	0.816	0.814
income/spatial_TX	0.80	0.869	0.870	0.872	0.872
income/temporal_CA14	0.99	0.905	0.904	0.904	0.904
mobility/spatial_MS	0.50	0.909	0.901	0.902	0.904
mobility/spatial_NY	0.79	0.923	0.925	0.923	0.926
mobility/spatial_PR	0.01	0.920	0.932	0.911	0.900
mobility/spatial_TX	0.72	0.883	0.891	0.894	0.894
mobility/temporal_CA14	0.92	0.879	0.880	0.881	0.880
pubcov/spatial_MS	0.80	0.921	1.000	0.921	1.000
pubcov/spatial_NY	0.97	0.889	1.000	0.897	1.000
pubcov/spatial_PR	0.80	0.860	0.954	0.894	1.000
pubcov/spatial_TX	0.94	0.938	0.977	0.938	1.000
pubcov/temporal_CA14	0.96	0.926	0.926	0.926	0.927
traveltime/spatial_MS	0.19	0.774	1.000	0.896	1.000
traveltime/spatial_NY	0.50	0.834	1.000	0.877	1.000
traveltime/spatial_PR	0.15	0.693	1.000	0.894	1.000
traveltime/spatial_TX	0.76	0.698	0.811	0.878	1.000
traveltime/temporal_CA14	0.95	0.901	0.903	0.902	0.903

Table 21: **Few-shot target calibration on ACS** (five tasks, 3 seeds). *Left:* coverage at nominal 0.90; at k=500 also the validity rate. *Right:* worst-class coverage, class-conditional (cc) against marginal few-shot calibration on the same labels.

method	TabICL	TabPFN
plain (src)	0.888	0.877
AUTO-τ (cov. wt.)	0.910	0.913
few k=50	0.900	0.901
few k=200	0.899	0.901
few k=500	0.898 / 0.92	0.899 / 0.91

k	method	TabICL	TabPFN
50	marginal	0.796	0.789
50	cc	**0.892**	**0.893**
200	marginal	0.784	0.783
200	cc	**0.882**	**0.882**
500	marginal	0.781	0.778
500	cc	**0.885**	**0.885**

Table 22: TabICLv2.

task	plain	$k50$	$k100$	$k200$	$k500$
income	0.862	0.901	0.900	0.905	0.898
employment	0.900	0.883	0.884	0.894	0.896
pubcov	0.924	0.900	0.903	0.898	0.901
mobility	0.908	0.908	0.901	0.898	0.896
traveltime	0.847	0.906	0.894	0.900	0.899

Table 23: TabPFNv2.

task	plain	$k50$	$k100$	$k200$	$k500$
income	0.857	0.917	0.911	0.910	0.900
employment	0.896	0.890	0.890	0.899	0.897
pubcov	0.918	0.897	0.899	0.895	0.901
mobility	0.903	0.902	0.895	0.896	0.895
traveltime	0.810	0.902	0.895	0.906	0.902

Table 24: **Every rule that uses the source, at k=50.** Tail (10th-percentile, the fifth-smallest of 46 settings), 25th-percentile and mean coverage, and mean set size, 5 seeds; WCC-CSPD uses per-class domain classifiers for its weights. Ungated pooling is below naive on every statistic, the two gated rules are above it, and the conservative form is certified but wider.

	TabICLv2				TabPFNv2			
rule	worst	p25	mean	size	worst	p25	mean	size
source-only	0.853	0.869	0.883	1.37	0.813	0.865	0.874	1.34
naive	0.879	0.887	0.900	1.42	0.878	0.888	0.898	1.40
pooled	0.864	0.884	0.897	1.41	0.818	0.879	0.882	1.36
pooled-DR	0.858	0.874	0.872	1.34	0.830	0.872	0.869	1.33
WCC-CSPD	0.844	0.864	0.873	1.39	0.841	0.865	0.872	1.39
NexCP-gated	**0.898**	0.906	0.914	1.46	**0.897**	0.902	0.911	1.44
SASC	**0.892**	0.904	0.917	1.46	**0.889**	0.899	0.912	1.44
SASC-cons	0.945	0.957	0.964	1.66	0.947	0.955	0.962	1.65

Table 25: **Tail reliability against label budget.** Tail (10th-percentile) coverage across the 46 concept-shift settings (5 ACS tasks $\times$ 9 targets + Wine), 5 seeds, nominal 0.90. Both gated rules at k=50 match or exceed naive at k=200–500.

tail cov.	k=20	k=30	k=50	k=100	k=200	k=500
naive (TabICL)	0.868	0.867	0.879	0.883	0.890	0.891
NexCP-gated (TabICL)	0.888	0.893	0.898	0.896	0.898	0.896
SASC (TabICL)	0.878	0.886	0.892	0.896	0.897	0.898
naive (TabPFN)	0.870	0.877	0.878	0.882	0.888	0.891
NexCP-gated (TabPFN)	0.891	0.894	0.897	0.897	0.897	0.896
SASC (TabPFN)	0.866	0.881	0.889	0.897	0.899	0.897

Table 26: Tail (10th-percentile) coverage across the 46 concept-shift settings, 5 seeds, $n_s = 3000$, for every calibration rule including the pooling baselines and the closest prior method. SASC $>$ naive at every budget for two of the five base models, TabICLv2 and GAMformer; the exceptions are TabPFNv2, TabPFNv3 and TabDPT at k=20, deficits of 0.003–0.008 that the paired test does not distinguish from zero (§X), and SASC-cons is above both throughout by construction. Nominal coverage is 0.90.

method	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2						
source-only	0.853	0.853	0.853	0.853	0.853	0.853
naive	0.868	0.867	0.879	0.883	0.890	0.891
pooled	0.832	0.853	0.864	0.871	0.867	0.869
pooled (density-ratio wt.)	0.858	0.858	0.858	0.859	0.861	0.865
WCC-CSPD	0.847	0.848	0.849	0.851	0.855	0.861
NexCP-gated	0.888	0.893	0.898	0.896	0.898	0.896
anchor only	0.832	0.853	0.863	0.869	0.863	0.864
anchor (loc+scale)	0.816	0.833	0.847	0.867	0.873	0.874
SASC (fixed w)	0.852	0.867	0.878	0.881	0.888	0.891
SASC	0.878	0.886	0.892	0.896	0.897	0.898
naive-cons	1.000	0.948	0.943	0.925	0.915	0.912
SASC-cons	1.000	0.950	0.945	0.935	0.922	0.914
TabPFNv2						
source-only	0.813	0.813	0.813	0.813	0.813	0.813
naive	0.870	0.877	0.878	0.882	0.888	0.891
pooled	0.810	0.819	0.818	0.792	0.796	0.813
pooled (density-ratio wt.)	0.829	0.830	0.830	0.832	0.835	0.844
WCC-CSPD	0.846	0.847	0.848	0.849	0.852	0.861
NexCP-gated	0.891	0.894	0.897	0.897	0.897	0.896
anchor only	0.808	0.818	0.818	0.785	0.785	0.787
anchor (loc+scale)	0.819	0.832	0.840	0.853	0.868	0.874
SASC (fixed w)	0.832	0.837	0.855	0.863	0.879	0.888
SASC	0.866	0.881	0.889	0.897	0.899	0.897
naive-cons	1.000	0.951	0.947	0.929	0.915	0.910
SASC-cons	1.000	0.952	0.947	0.933	0.919	0.912
TabPFNv3						
source-only	0.814	0.814	0.814	0.814	0.814	0.814
naive	0.867	0.870	0.879	0.883	0.888	0.894
pooled	0.761	0.781	0.779	0.771	0.775	0.795
pooled (density-ratio wt.)	0.820	0.820	0.822	0.825	0.829	0.839
WCC-CSPD	0.842	0.842	0.844	0.846	0.848	0.857
NexCP-gated	0.891	0.890	0.894	0.896	0.896	0.896
anchor only	0.759	0.779	0.775	0.761	0.759	0.755
anchor (loc+scale)	0.828	0.835	0.838	0.862	0.874	0.874
SASC (fixed w)	0.806	0.828	0.833	0.851	0.877	0.887
SASC	0.863	0.881	0.890	0.897	0.894	0.896
naive-cons	1.000	0.949	0.945	0.926	0.913	0.912
SASC-cons	1.000	0.952	0.946	0.929	0.919	0.913
TabDPT						
source-only	0.792	0.792	0.792	0.792	0.792	0.792
naive	0.864	0.870	0.878	0.882	0.889	0.890
pooled	0.780	0.784	0.768	0.765	0.763	0.784
pooled (density-ratio wt.)	0.814	0.815	0.817	0.820	0.826	0.834
WCC-CSPD	0.842	0.844	0.846	0.848	0.852	0.860
NexCP-gated	0.892	0.886	0.895	0.891	0.894	0.895
anchor only	0.780	0.782	0.766	0.761	0.756	0.754
anchor (loc+scale)	0.809	0.839	0.842	0.858	0.865	0.868
SASC (fixed w)	0.813	0.827	0.828	0.850	0.869	0.886
SASC	0.856	0.879	0.887	0.895	0.894	0.895
naive-cons	1.000	0.950	0.944	0.925	0.915	0.910

continued on the next page

method	k=20	k=30	k=50	k=100	k=200	k=500
SASC-cons	1.000	0.953	0.944	0.929	0.919	0.911
GAMformer						
source-only	0.867	0.867	0.867	0.867	0.867	0.867
naive	0.865	0.877	0.878	0.884	0.889	0.891
pooled	0.887	0.879	0.888	0.886	0.887	0.889
pooled (density-ratio wt.)	0.870	0.870	0.870	0.870	0.873	0.878
WCC-CSPD	0.859	0.859	0.859	0.857	0.862	0.869
NexCP-gated	0.892	0.894	0.896	0.901	0.896	0.895
anchor only	0.887	0.880	0.888	0.884	0.886	0.890
anchor (loc+scale)	0.805	0.798	0.810	0.811	0.815	0.821
SASC (fixed w)	0.884	0.887	0.885	0.889	0.891	0.892
SASC	0.893	0.898	0.898	0.901	0.899	0.900
naive-cons	1.000	0.949	0.946	0.929	0.918	0.909
SASC-cons	1.000	0.953	0.949	0.935	0.921	0.914
XGBoost						
source-only	0.858	0.858	0.858	0.858	0.858	0.858
naive	0.856	0.862	0.874	0.884	0.888	0.890
pooled	0.877	0.870	0.877	0.875	0.880	0.885
pooled (density-ratio wt.)	0.861	0.862	0.862	0.862	0.864	0.868
WCC-CSPD	0.860	0.861	0.861	0.863	0.865	0.872
NexCP-gated	0.882	0.890	0.892	0.893	0.890	0.896
anchor only	0.876	0.870	0.877	0.874	0.880	0.883
anchor (loc+scale)	0.811	0.829	0.844	0.870	0.869	0.882
SASC (fixed w)	0.866	0.875	0.881	0.887	0.889	0.890
SASC	0.888	0.892	0.895	0.894	0.896	0.898
naive-cons	1.000	0.948	0.943	0.928	0.916	0.909
SASC-cons	1.000	0.948	0.943	0.930	0.917	0.914
Logistic reg.						
source-only	0.870	0.870	0.870	0.870	0.870	0.870
naive	0.851	0.865	0.872	0.877	0.885	0.890
pooled	0.877	0.877	0.882	0.885	0.884	0.881
pooled (density-ratio wt.)	0.872	0.872	0.872	0.873	0.874	0.878
WCC-CSPD	0.874	0.874	0.875	0.876	0.878	0.880
NexCP-gated	0.895	0.892	0.896	0.895	0.896	0.896
anchor only	0.876	0.877	0.883	0.885	0.883	0.877
anchor (loc+scale)	0.799	0.825	0.832	0.838	0.837	0.842
SASC (fixed w)	0.873	0.879	0.884	0.885	0.886	0.890
SASC	0.882	0.895	0.894	0.899	0.898	0.899
naive-cons	1.000	0.954	0.950	0.928	0.915	0.909
SASC-cons	1.000	0.954	0.952	0.930	0.919	0.913

Table 27: Mean coverage over the same suite. SASC does *not* improve mean coverage, since it mildly over-covers, which is why we report the tail as the headline. Nominal coverage is 0.90.

method	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2						
source-only	0.883	0.883	0.883	0.883	0.883	0.883
naive	0.898	0.899	0.900	0.900	0.900	0.900
pooled	0.893	0.894	0.897	0.900	0.897	0.899
pooled (density-ratio wt.)	0.871	0.871	0.872	0.878	0.884	0.889
WCC-CSPD	0.883	0.883	0.883	0.884	0.886	0.889
NexCP-gated	0.914	0.913	0.914	0.912	0.910	0.908
anchor only	0.893	0.894	0.896	0.900	0.897	0.898
anchor (loc+scale)	0.870	0.880	0.889	0.905	0.908	0.910
SASC (fixed w)	0.897	0.898	0.900	0.902	0.901	0.901
SASC	0.909	0.912	0.917	0.919	0.916	0.915

continued on the next page

method	k=20	k=30	k=50	k=100	k=200	k=500
naive-cons	1.000	0.967	0.961	0.941	0.925	0.919
SASC-cons	1.000	0.970	0.964	0.948	0.933	0.927
TabPFNv2						
source-only	0.874	0.874	0.874	0.874	0.874	0.874
naive	0.900	0.900	0.898	0.900	0.901	0.901
pooled	0.880	0.883	0.882	0.883	0.882	0.887
pooled (density-ratio wt.)	0.868	0.868	0.869	0.874	0.880	0.885
WCC-CSPD	0.884	0.884	0.884	0.885	0.886	0.890
NexCP-gated	0.912	0.912	0.911	0.911	0.908	0.907
anchor only	0.880	0.882	0.882	0.882	0.881	0.882
anchor (loc+scale)	0.872	0.886	0.894	0.905	0.910	0.913
SASC (fixed w)	0.887	0.890	0.891	0.895	0.897	0.900
SASC	0.902	0.909	0.912	0.915	0.912	0.912
naive-cons	1.000	0.967	0.961	0.941	0.925	0.919
SASC-cons	1.000	0.968	0.962	0.944	0.930	0.924
TabPFNv3						
source-only	0.871	0.871	0.871	0.871	0.871	0.871
naive	0.900	0.899	0.898	0.901	0.900	0.901
pooled	0.877	0.880	0.879	0.882	0.881	0.885
pooled (density-ratio wt.)	0.863	0.863	0.864	0.870	0.877	0.884
WCC-CSPD	0.880	0.880	0.880	0.881	0.883	0.888
NexCP-gated	0.910	0.911	0.911	0.910	0.907	0.906
anchor only	0.877	0.880	0.878	0.881	0.879	0.880
anchor (loc+scale)	0.870	0.891	0.895	0.908	0.913	0.916
SASC (fixed w)	0.884	0.887	0.888	0.895	0.896	0.900
SASC	0.900	0.908	0.912	0.916	0.912	0.912
naive-cons	1.000	0.967	0.960	0.941	0.925	0.919
SASC-cons	1.000	0.969	0.962	0.945	0.930	0.924
TabDPT						
source-only	0.872	0.872	0.872	0.872	0.872	0.872
naive	0.902	0.900	0.899	0.900	0.900	0.901
pooled	0.880	0.879	0.878	0.880	0.878	0.882
pooled (density-ratio wt.)	0.864	0.865	0.865	0.871	0.877	0.883
WCC-CSPD	0.878	0.878	0.879	0.880	0.882	0.886
NexCP-gated	0.911	0.911	0.911	0.910	0.908	0.906
anchor only	0.880	0.879	0.878	0.879	0.877	0.878
anchor (loc+scale)	0.865	0.883	0.890	0.904	0.909	0.913
SASC (fixed w)	0.887	0.887	0.888	0.893	0.896	0.899
SASC	0.903	0.908	0.912	0.915	0.912	0.911
naive-cons	1.000	0.966	0.959	0.941	0.926	0.918
SASC-cons	1.000	0.968	0.961	0.945	0.930	0.924
GAMformer						
source-only	0.899	0.899	0.899	0.899	0.899	0.899
naive	0.903	0.899	0.898	0.901	0.902	0.900
pooled	0.911	0.913	0.913	0.914	0.915	0.914
pooled (density-ratio wt.)	0.892	0.892	0.892	0.893	0.896	0.899
WCC-CSPD	0.893	0.892	0.893	0.893	0.894	0.896
NexCP-gated	0.919	0.915	0.915	0.914	0.913	0.910
anchor only	0.911	0.913	0.913	0.915	0.916	0.915
anchor (loc+scale)	0.873	0.882	0.890	0.902	0.906	0.914
SASC (fixed w)	0.912	0.912	0.909	0.909	0.908	0.903
SASC	0.919	0.923	0.924	0.925	0.925	0.923
naive-cons	1.000	0.966	0.961	0.941	0.926	0.918
SASC-cons	1.000	0.970	0.965	0.949	0.939	0.932
XGBoost						
source-only	0.887	0.887	0.887	0.887	0.887	0.887
naive	0.898	0.900	0.899	0.902	0.901	0.902
pooled	0.903	0.902	0.903	0.904	0.902	0.903
pooled (density-ratio wt.)	0.876	0.876	0.876	0.880	0.885	0.889

continued on the next page

method	k=20	k=30	k=50	k=100	k=200	k=500
WCC-CSPD	0.887	0.886	0.886	0.887	0.888	0.890
NexCP-gated	0.910	0.911	0.909	0.909	0.907	0.906
anchor only	0.903	0.902	0.903	0.904	0.903	0.904
anchor (loc+scale)	0.876	0.886	0.902	0.915	0.919	0.920
SASC (fixed w)	0.903	0.902	0.902	0.903	0.902	0.903
SASC	0.915	0.916	0.918	0.918	0.915	0.915
naive-cons	1.000	0.967	0.961	0.940	0.926	0.920
SASC-cons	1.000	0.970	0.963	0.946	0.932	0.926
Logistic reg.						
source-only	0.895	0.895	0.895	0.895	0.895	0.895
naive	0.897	0.898	0.899	0.899	0.900	0.900
pooled	0.908	0.907	0.908	0.909	0.906	0.905
pooled (density-ratio wt.)	0.896	0.896	0.896	0.897	0.898	0.898
WCC-CSPD	0.894	0.895	0.895	0.895	0.895	0.897
NexCP-gated	0.916	0.916	0.916	0.913	0.911	0.908
anchor only	0.908	0.907	0.908	0.909	0.906	0.905
anchor (loc+scale)	0.862	0.874	0.886	0.895	0.902	0.907
SASC (fixed w)	0.906	0.905	0.904	0.902	0.901	0.900
SASC	0.914	0.916	0.916	0.917	0.914	0.912
naive-cons	1.000	0.969	0.961	0.941	0.926	0.917
SASC-cons	1.000	0.970	0.963	0.944	0.930	0.922

Table 28: Mean prediction-set size over the same suite (number of labels; 2 classes throughout the concept-shift family, so 1.0 is a singleton and 2.0 is the full label set).

method	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2						
source-only	1.365	1.365	1.365	1.365	1.365	1.365
naive	1.426	1.421	1.417	1.409	1.405	1.405
pooled	1.402	1.398	1.406	1.412	1.402	1.405
pooled (density-ratio wt.)	1.342	1.343	1.343	1.357	1.369	1.380
WCC-CSPD	1.415	1.414	1.415	1.416	1.419	1.426
NexCP-gated	1.461	1.454	1.456	1.443	1.435	1.429
anchor only	1.402	1.398	1.406	1.412	1.402	1.403
anchor (loc+scale)	1.386	1.403	1.407	1.449	1.451	1.446
SASC (fixed w)	1.409	1.406	1.411	1.411	1.406	1.406
SASC	1.446	1.448	1.464	1.465	1.452	1.449
naive-cons	2.000	1.699	1.652	1.557	1.490	1.465
SASC-cons	2.000	1.706	1.661	1.575	1.514	1.489
TabPFNv2						
source-only	1.335	1.335	1.335	1.335	1.335	1.335
naive	1.424	1.413	1.402	1.399	1.397	1.397
pooled	1.364	1.362	1.361	1.361	1.356	1.365
pooled (density-ratio wt.)	1.327	1.327	1.328	1.339	1.351	1.363
WCC-CSPD	1.418	1.419	1.419	1.421	1.423	1.429
NexCP-gated	1.448	1.441	1.438	1.433	1.424	1.419
anchor only	1.364	1.363	1.361	1.360	1.353	1.357
anchor (loc+scale)	1.376	1.403	1.407	1.433	1.435	1.437
SASC (fixed w)	1.376	1.378	1.379	1.386	1.388	1.393
SASC	1.421	1.434	1.440	1.445	1.435	1.432
naive-cons	2.000	1.690	1.642	1.545	1.480	1.456
SASC-cons	2.000	1.693	1.646	1.555	1.496	1.472
TabPFNv3						
source-only	1.325	1.325	1.325	1.325	1.325	1.325
naive	1.421	1.408	1.401	1.401	1.392	1.396
pooled	1.355	1.355	1.351	1.357	1.351	1.359

continued on the next page

method	k=20	k=30	k=50	k=100	k=200	k=500
pooled (density-ratio wt.)	1.313	1.314	1.315	1.327	1.341	1.356
WCC-CSPD	1.403	1.404	1.405	1.406	1.410	1.418
NexCP-gated	1.441	1.438	1.436	1.431	1.419	1.414
anchor only	1.355	1.354	1.350	1.355	1.347	1.348
anchor (loc+scale)	1.372	1.412	1.410	1.437	1.441	1.443
SASC (fixed w)	1.368	1.368	1.370	1.383	1.382	1.391
SASC	1.414	1.429	1.440	1.448	1.433	1.430
naive-cons	2.000	1.690	1.641	1.547	1.478	1.454
SASC-cons	2.000	1.695	1.649	1.560	1.496	1.471
TabDPT						
source-only	1.322	1.322	1.322	1.322	1.322	1.322
naive	1.426	1.412	1.400	1.397	1.394	1.394
pooled	1.359	1.350	1.347	1.349	1.345	1.351
pooled (density-ratio wt.)	1.312	1.312	1.313	1.324	1.335	1.350
WCC-CSPD	1.396	1.396	1.397	1.399	1.403	1.410
NexCP-gated	1.445	1.440	1.435	1.431	1.419	1.413
anchor only	1.359	1.350	1.347	1.349	1.343	1.344
anchor (loc+scale)	1.357	1.387	1.393	1.426	1.429	1.438
SASC (fixed w)	1.373	1.368	1.367	1.377	1.381	1.388
SASC	1.422	1.428	1.437	1.444	1.434	1.430
naive-cons	2.000	1.687	1.639	1.545	1.481	1.453
SASC-cons	2.000	1.692	1.645	1.557	1.497	1.471
GAMformer						
source-only	1.441	1.441	1.441	1.441	1.441	1.441
naive	1.474	1.453	1.440	1.443	1.444	1.438
pooled	1.488	1.485	1.482	1.484	1.483	1.479
pooled (density-ratio wt.)	1.426	1.426	1.425	1.427	1.433	1.440
WCC-CSPD	1.483	1.483	1.484	1.483	1.485	1.487
NexCP-gated	1.509	1.497	1.492	1.488	1.481	1.471
anchor only	1.489	1.486	1.484	1.486	1.486	1.484
anchor (loc+scale)	1.436	1.458	1.468	1.485	1.490	1.509
SASC (fixed w)	1.485	1.477	1.466	1.462	1.456	1.444
SASC	1.515	1.516	1.517	1.514	1.513	1.504
naive-cons	2.000	1.719	1.683	1.590	1.528	1.497
SASC-cons	2.000	1.733	1.695	1.611	1.562	1.534
XGBoost						
source-only	1.412	1.412	1.412	1.412	1.412	1.412
naive	1.464	1.463	1.451	1.454	1.450	1.451
pooled	1.478	1.466	1.461	1.462	1.454	1.453
pooled (density-ratio wt.)	1.389	1.389	1.390	1.398	1.408	1.416
WCC-CSPD	1.463	1.463	1.463	1.464	1.466	1.470
NexCP-gated	1.488	1.489	1.479	1.478	1.469	1.464
anchor only	1.479	1.466	1.461	1.463	1.456	1.456
anchor (loc+scale)	1.476	1.493	1.516	1.549	1.546	1.544
SASC (fixed w)	1.468	1.459	1.452	1.455	1.451	1.452
SASC	1.512	1.507	1.508	1.504	1.494	1.490
naive-cons	2.000	1.722	1.684	1.589	1.532	1.508
SASC-cons	2.000	1.734	1.694	1.609	1.553	1.529
Logistic reg.						
source-only	1.467	1.467	1.467	1.467	1.467	1.467
naive	1.489	1.492	1.475	1.469	1.471	1.470
pooled	1.515	1.507	1.505	1.505	1.492	1.487
pooled (density-ratio wt.)	1.465	1.465	1.465	1.467	1.469	1.470
WCC-CSPD	1.532	1.532	1.533	1.533	1.534	1.535
NexCP-gated	1.539	1.539	1.533	1.521	1.514	1.505
anchor only	1.515	1.507	1.505	1.507	1.494	1.490
anchor (loc+scale)	1.406	1.428	1.458	1.473	1.487	1.507
SASC (fixed w)	1.506	1.500	1.486	1.477	1.474	1.471
SASC	1.535	1.540	1.532	1.533	1.518	1.514
naive-cons	2.000	1.767	1.725	1.630	1.563	1.531

continued on the next page

method	k=20	k=30	k=50	k=100	k=200	k=500
SASC-cons	2.000	1.772	1.730	1.642	1.581	1.550

M FULL PER-SETTING RESULTS (46 SETTINGS)

Tables 29–32 give the full per-setting coverage of naive few-shot conformal and SASC for every label budget and every base model. Figure 13 (ranked setting by setting), Figure 14 (paired, method against method) and Figure 15 (the full coverage distributions) show the same data three ways. The main text summarizes these numbers through its 10th-percentile statistic. We include them so a reader can audit any claim, in particular that SASC's advantage is concentrated in the hardest settings rather than spread uniformly, and that it gives up ground only where naive already over-covers.

Three patterns are visible directly in the tables. *(i) The gain lands where it is needed.* Splitting the settings at the median of naive's coverage, SASC's mean improvement at k=50 is $+0.024$ in the harder half and $+0.010$ in the easier half. The ratio is about two and a half, though the gain is not zero in the easy half. *(ii) On a near-null shift the method does no harm, but it does not switch off.* The CA-2014 temporal control is barely shifted: naive already attains 0.905 there, and SASC moves it to 0.913. The mean selected weight there ($\bar{w} = 0.29$) is close to the global average (0.30), so the gate is not refusing the anchor. What protects the method is that on a near-null shift the anchor is *accurate* and merely unnecessary, and leaning on an accurate anchor costs only mild over-coverage. The real switch-off appears where the anchor is *biased*, which is the 50k-scale regime of §G rather than the mild-shift regime. *(iii) The hardest settings are defined by task more than by geography.* For TabICL the five lowest-coverage settings under naive calibration are public-coverage→PR, public-coverage→WV, employment→NY, travel-time→CA-2014 and public-coverage→OK. Public coverage is a task whose conditional depends directly on state-level policy, so it shifts hardest wherever it goes. Puerto Rico is the single most-shifted *target* (its anchor bias $|b| = 0.053$ is well above the 0.030 average). SASC still improves it ($0.895 \to 0.926$ at k=50), because the CV gate admits a partial anchor rather than forcing an all-or-nothing choice.

Table 29: Per-setting coverage, naive vs SASC, TabICLv2 (5 seeds, nominal 0.90). Bold marks the larger of the two.

setting	k=20		k=50		k=100		k=200		k=500	
	NV	SASC	NV	SASC	NV	SASC	NV	SASC	NV	SASC
employment_spatial_FL	0.920	**0.924**	0.891	**0.911**	0.892	**0.910**	0.898	**0.912**	**0.911**	0.911
employment_spatial_MS	0.910	**0.922**	**0.925**	0.924	0.911	**0.913**	0.913	**0.913**	0.900	**0.903**
employment_spatial_NY	0.868	**0.901**	0.871	**0.890**	0.890	**0.906**	0.899	**0.908**	0.908	**0.912**
employment_spatial_OK	0.875	**0.907**	0.908	**0.917**	**0.929**	0.913	0.917	**0.918**	**0.906**	0.906
employment_spatial_PR	**0.918**	0.876	**0.891**	0.883	0.880	**0.886**	0.889	**0.891**	0.889	**0.889**
employment_spatial_TX	**0.915**	0.910	**0.914**	0.909	**0.917**	0.913	0.895	**0.897**	0.891	**0.894**
employment_spatial_WA	0.855	**0.896**	0.881	**0.900**	0.896	**0.906**	0.900	**0.909**	0.908	**0.913**
employment_spatial_WV	**0.925**	0.915	0.906	**0.915**	0.916	**0.928**	**0.922**	0.922	0.914	**0.914**
employment_temporal_CA14	**0.923**	0.901	**0.934**	0.924	0.920	**0.921**	**0.904**	0.903	0.900	**0.900**
income_spatial_FL	0.899	**0.912**	**0.887**	0.886	0.895	**0.899**	0.886	**0.902**	0.903	**0.908**
income_spatial_MS	**0.926**	0.910	0.914	**0.919**	0.907	**0.910**	**0.915**	0.911	0.909	**0.910**
income_spatial_NY	0.853	**0.885**	0.882	**0.884**	0.892	**0.899**	0.893	**0.897**	**0.900**	0.899
income_spatial_OK	0.880	**0.885**	0.890	**0.891**	0.882	**0.894**	0.892	**0.903**	0.901	**0.907**
income_spatial_PR	**0.903**	0.879	**0.899**	0.894	**0.914**	0.909	**0.907**	0.904	**0.893**	0.893
income_spatial_TX	**0.906**	0.901	0.906	**0.914**	0.896	**0.908**	0.897	**0.900**	0.890	**0.896**
income_spatial_WA	0.917	**0.938**	0.906	**0.915**	0.902	**0.913**	0.895	**0.911**	0.897	**0.908**
income_spatial_WV	0.847	**0.901**	0.912	**0.924**	0.904	**0.918**	0.896	**0.906**	0.893	**0.901**
income_temporal_CA14	0.920	**0.930**	0.903	**0.919**	0.910	**0.921**	0.902	**0.907**	0.890	**0.906**
mobility_spatial_FL	0.872	**0.903**	0.893	**0.906**	0.883	**0.894**	0.897	**0.898**	**0.903**	0.902
mobility_spatial_MS	0.911	**0.939**	0.887	**0.930**	0.906	**0.928**	0.902	**0.924**	0.904	**0.920**
mobility_spatial_NY	0.892	**0.912**	0.885	**0.916**	0.897	**0.926**	0.893	**0.923**	0.892	**0.924**
mobility_spatial_OK	0.875	**0.918**	0.893	**0.925**	0.907	**0.944**	0.913	**0.944**	0.911	**0.944**
mobility_spatial_PR	**0.934**	0.918	0.915	**0.926**	0.887	**0.907**	0.890	**0.904**	0.895	**0.907**
mobility_spatial_TX	**0.934**	0.929	0.910	**0.916**	0.909	**0.921**	0.907	**0.923**	0.900	**0.915**
mobility_spatial_WA	0.870	**0.918**	0.921	**0.940**	0.917	**0.954**	0.924	**0.938**	0.915	**0.934**
mobility_spatial_WV	0.909	**0.961**	0.909	**0.968**	0.902	**0.952**	0.897	**0.943**	0.897	**0.941**
mobility_temporal_CA14	**0.950**	0.936	**0.924**	0.909	**0.918**	0.911	0.913	**0.916**	**0.915**	0.914
pubcov_spatial_FL	0.917	**0.938**	0.924	**0.930**	0.911	**0.920**	0.893	**0.911**	0.900	**0.902**
pubcov_spatial_MS	**0.917**	0.914	0.881	**0.895**	0.873	**0.898**	0.876	**0.899**	0.892	**0.905**
pubcov_spatial_NY	**0.868**	0.836	**0.912**	0.910	0.891	**0.903**	0.898	**0.902**	**0.895**	0.894
pubcov_spatial_OK	0.818	**0.857**	0.878	**0.900**	0.892	**0.901**	**0.898**	0.897	0.899	**0.900**
pubcov_spatial_PR	0.884	**0.941**	0.865	**0.959**	0.876	**0.959**	0.893	**0.956**	0.903	**0.962**

setting	k=20		k=50		k=100		k=200		k=500	
	nv	sasc	nv	sasc	nv	sasc	nv	sasc	nv	sasc
pubcov_spatial_TX	**0.926**	0.919	0.907	**0.920**	0.891	**0.917**	0.897	**0.919**	0.901	**0.915**
pubcov_spatial_WA	**0.900**	0.890	**0.912**	0.909	**0.910**	0.908	**0.900**	0.896	**0.900**	0.899
pubcov_spatial_WV	0.873	**0.907**	0.869	**0.900**	0.893	**0.895**	**0.901**	0.898	0.899	**0.899**
pubcov_temporal_CA14	**0.886**	0.882	0.893	**0.921**	0.909	**0.932**	0.906	**0.927**	0.910	**0.929**
traveltime_spatial_FL	**0.895**	0.832	0.910	**0.919**	0.907	**0.923**	0.897	**0.908**	0.896	**0.916**
traveltime_spatial_MS	0.884	**0.903**	0.885	**0.924**	0.895	**0.928**	0.894	**0.927**	0.887	**0.924**
traveltime_spatial_NY	**0.892**	0.880	**0.911**	0.903	0.900	**0.905**	0.895	**0.908**	0.896	**0.907**
traveltime_spatial_OK	0.895	**0.937**	0.913	**0.939**	0.886	**0.941**	0.891	**0.932**	0.898	**0.944**
traveltime_spatial_PR	0.915	**0.983**	0.908	**0.968**	0.899	**0.950**	0.888	**0.943**	0.893	**0.949**
traveltime_spatial_TX	0.880	**0.906**	0.907	**0.926**	0.908	**0.924**	0.906	**0.929**	0.905	**0.927**
traveltime_spatial_WA	0.901	**0.910**	0.898	**0.907**	0.911	**0.936**	0.903	**0.923**	0.908	**0.928**
traveltime_spatial_WV	**0.936**	0.901	0.887	**0.917**	0.885	**0.919**	0.900	**0.923**	0.890	**0.921**
traveltime_temporal_CA14	**0.884**	0.868	0.871	**0.892**	0.870	**0.892**	**0.897**	0.895	0.898	**0.911**
wine_r2w	0.927	**0.982**	0.898	**0.979**	0.896	**1.000**	0.904	**0.998**	0.904	**0.997**
mean	0.898	0.909	0.900	0.917	0.900	0.919	0.900	0.916	0.900	0.915
10th pctile	0.868	0.878	0.879	0.892	0.883	0.896	0.890	0.897	0.891	0.898

Table 30: Per-setting coverage, naive vs sasc, TabPFNv2 (5 seeds, nominal 0.90). Bold marks the larger of the two.

setting	k=20		k=50		k=100		k=200		k=500	
	nv	sasc	nv	sasc	nv	sasc	nv	sasc	nv	sasc
employment_spatial_FL	0.917	**0.926**	0.901	**0.916**	0.899	**0.911**	0.907	**0.915**	0.911	**0.913**
employment_spatial_MS	0.910	**0.923**	**0.921**	0.913	**0.913**	0.911	**0.914**	0.911	0.904	**0.905**
employment_spatial_NY	0.871	**0.915**	0.873	**0.898**	0.884	**0.904**	0.906	**0.911**	0.910	**0.914**
employment_spatial_OK	0.868	**0.905**	0.919	**0.920**	**0.921**	0.911	**0.922**	0.920	0.904	**0.905**
employment_spatial_PR	**0.920**	0.881	0.881	**0.887**	0.855	**0.883**	0.880	**0.884**	**0.889**	0.889
employment_spatial_TX	**0.922**	0.911	**0.922**	0.917	**0.930**	0.927	0.899	**0.900**	**0.897**	0.897
employment_spatial_WA	0.860	**0.891**	0.892	**0.893**	0.898	**0.904**	0.904	**0.909**	0.905	**0.911**
employment_spatial_WV	**0.933**	0.907	0.913	**0.914**	0.919	**0.927**	**0.921**	0.918	**0.913**	0.912
employment_temporal_CA14	**0.908**	0.901	**0.933**	0.925	0.913	**0.913**	0.906	**0.908**	**0.901**	0.900
income_spatial_FL	0.890	**0.916**	0.890	**0.898**	0.884	**0.898**	0.887	**0.901**	0.902	**0.908**
income_spatial_MS	**0.936**	0.915	0.900	**0.901**	0.903	**0.914**	0.908	**0.912**	0.910	**0.912**
income_spatial_NY	0.853	**0.882**	0.871	**0.881**	**0.913**	0.906	0.905	**0.905**	0.907	**0.909**
income_spatial_OK	0.872	**0.894**	0.888	**0.904**	0.892	**0.906**	0.905	**0.906**	0.902	**0.906**
income_spatial_PR	**0.907**	0.866	0.901	**0.902**	**0.915**	0.915	0.900	**0.903**	**0.891**	0.889
income_spatial_TX	0.904	**0.912**	0.901	**0.910**	0.893	**0.902**	0.888	**0.898**	0.885	**0.895**
income_spatial_WA	0.916	**0.934**	0.900	**0.913**	0.910	**0.919**	0.904	**0.912**	0.900	**0.909**
income_spatial_WV	**0.902**	0.901	0.898	**0.914**	0.894	**0.913**	0.894	**0.900**	0.893	**0.903**
income_temporal_CA14	0.902	**0.914**	0.881	**0.908**	0.909	**0.912**	0.896	**0.903**	0.896	**0.906**
mobility_spatial_FL	0.880	**0.911**	0.891	**0.913**	0.881	**0.895**	0.893	**0.895**	0.898	**0.899**
mobility_spatial_MS	0.916	**0.937**	0.892	**0.927**	0.912	**0.930**	0.903	**0.923**	0.904	**0.918**
mobility_spatial_NY	0.895	**0.915**	0.886	**0.913**	0.897	**0.924**	0.891	**0.922**	0.892	**0.924**
mobility_spatial_OK	0.879	**0.905**	0.897	**0.923**	0.910	**0.941**	0.916	**0.937**	0.910	**0.935**
mobility_spatial_PR	**0.943**	0.914	0.908	**0.922**	0.887	**0.905**	0.895	**0.903**	0.896	**0.906**
mobility_spatial_TX	0.931	**0.933**	**0.915**	0.913	0.909	**0.916**	0.916	**0.920**	0.905	**0.917**
mobility_spatial_WA	0.872	**0.918**	0.924	**0.936**	0.921	**0.949**	0.918	**0.939**	0.913	**0.930**
mobility_spatial_WV	0.912	**0.958**	0.907	**0.954**	0.898	**0.949**	0.894	**0.940**	0.896	**0.939**
mobility_temporal_CA14	**0.947**	0.941	0.907	**0.924**	**0.922**	0.910	**0.920**	0.919	0.912	0.912
pubcov_spatial_FL	0.917	**0.933**	0.921	**0.935**	0.898	**0.921**	0.890	**0.910**	0.894	**0.900**
pubcov_spatial_MS	0.908	**0.923**	0.883	**0.920**	0.871	**0.917**	0.877	**0.917**	0.894	**0.916**
pubcov_spatial_NY	**0.886**	0.850	0.892	**0.892**	0.894	**0.904**	**0.899**	0.898	**0.891**	0.889
pubcov_spatial_OK	0.839	**0.884**	0.872	**0.894**	0.895	**0.899**	0.900	**0.900**	0.898	**0.902**
pubcov_spatial_PR	0.888	**0.924**	0.888	**0.951**	0.885	**0.951**	0.891	**0.958**	0.902	**0.958**
pubcov_spatial_TX	0.901	**0.918**	0.880	**0.917**	0.878	**0.913**	0.877	**0.917**	0.896	**0.921**
pubcov_spatial_WA	**0.905**	0.903	0.901	**0.906**	**0.913**	0.909	**0.902**	0.901	**0.900**	0.899
pubcov_spatial_WV	0.885	**0.890**	0.876	**0.889**	0.890	**0.892**	**0.887**	0.886	**0.898**	0.897
pubcov_temporal_CA14	0.857	**0.899**	0.895	**0.918**	0.886	**0.929**	0.897	**0.929**	0.904	**0.931**
traveltime_spatial_FL	**0.890**	0.804	0.885	**0.885**	0.905	0.905	0.900	0.900	0.910	0.910
traveltime_spatial_MS	**0.918**	0.899	0.910	**0.915**	0.897	**0.911**	0.895	**0.905**	0.891	**0.904**
traveltime_spatial_NY	**0.889**	0.867	**0.906**	0.902	**0.910**	0.908	**0.907**	0.905	0.901	0.901
traveltime_spatial_OK	**0.875**	0.865	0.905	**0.914**	0.886	**0.915**	0.891	**0.914**	0.908	**0.933**
traveltime_spatial_PR	**0.928**	0.898	0.914	**0.959**	0.895	**0.949**	0.896	**0.950**	0.900	**0.958**
traveltime_spatial_TX	**0.911**	0.871	**0.897**	0.890	**0.906**	0.904	**0.901**	0.899	0.911	0.911
traveltime_spatial_WA	**0.923**	0.901	**0.920**	0.903	0.933	0.933	**0.922**	0.920	0.917	0.917
traveltime_spatial_WV	**0.914**	0.836	0.876	**0.890**	0.888	**0.893**	0.893	**0.903**	0.884	**0.899**
traveltime_temporal_CA14	**0.899**	0.869	**0.888**	0.880	0.881	**0.893**	**0.911**	0.910	0.904	**0.906**
wine_r2w	0.911	**0.937**	0.906	**0.937**	0.906	**0.937**	0.898	**0.932**	0.906	**0.945**
mean	0.900	0.902	0.898	0.912	0.900	0.915	0.901	0.912	0.901	0.912
10th pctile	0.870	0.866	0.878	0.889	0.882	0.897	0.888	0.899	0.891	0.897

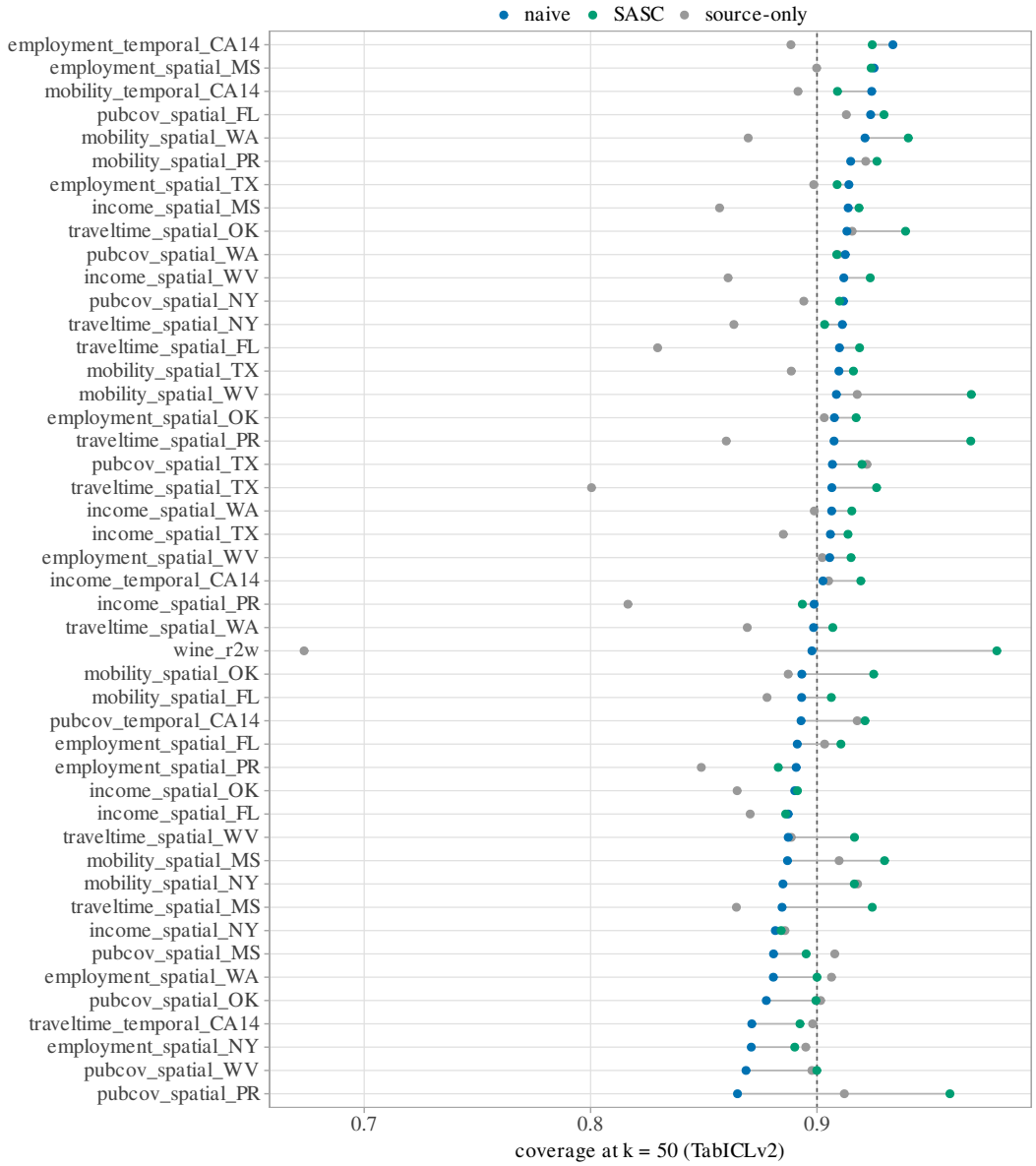


Figure 13: **All** 46 **settings at** k=50**, ranked by how hard they are for naive few-shot conformal** (TabICLv2). Gray: source-only calibration, free to a practitioner and scattered far below nominal. Blue: naive few-shot. Green: SASC. The gray–blue gap is what 50 target labels buy; the blue–green gap is what SASC adds on top, and it is visibly larger in the lower part of the ranking. Wine (`wine_r2w`) is the extreme case: source-only attains 0.67, naive 0.90, SASC 0.98.

Table 31: Per-setting coverage, naive vs SASC, XGBoost (5 seeds, nominal 0.90). Bold marks the larger of the two.

setting	k=20		k=50		k=100		k=200		k=500	
	NV	SASC	NV	SASC	NV	SASC	NV	SASC	NV	SASC
employment_spatial_FL	0.914	**0.927**	0.893	**0.917**	0.890	**0.908**	0.902	**0.919**	0.907	**0.912**
employment_spatial_MS	**0.933**	0.919	**0.922**	0.919	0.906	**0.915**	0.918	**0.923**	0.910	**0.917**
employment_spatial_NY	0.898	**0.904**	0.885	**0.909**	0.892	**0.898**	0.892	**0.909**	0.908	**0.910**
employment_spatial_OK	0.892	**0.901**	0.899	**0.917**	0.912	**0.913**	**0.921**	0.918	0.913	**0.916**
employment_spatial_PR	0.870	**0.881**	0.854	**0.874**	**0.874**	0.872	0.873	**0.879**	0.889	**0.892**
employment_spatial_TX	**0.932**	0.932	**0.929**	0.916	**0.926**	0.925	0.900	**0.901**	0.903	**0.905**
employment_spatial_WA	0.888	**0.897**	**0.920**	0.912	0.910	**0.914**	0.914	**0.920**	0.911	**0.917**
employment_spatial_WV	**0.939**	0.934	0.907	**0.912**	0.908	**0.915**	0.915	**0.919**	0.911	**0.917**
employment_temporal_CA14	**0.934**	0.908	0.929	**0.929**	**0.921**	0.915	**0.906**	0.906	**0.903**	0.902

setting	k=20		k=50		k=100		k=200		k=500	
	nv	SASC	nv	SASC	nv	SASC	nv	SASC	nv	SASC
income_spatial_FL	0.857	**0.902**	0.873	**0.895**	**0.898**	0.895	**0.894**	0.890	0.902	**0.904**
income_spatial_MS	**0.935**	0.926	0.918	**0.923**	0.912	**0.926**	**0.913**	0.910	**0.907**	0.905
income_spatial_NY	0.850	**0.895**	0.848	**0.888**	**0.889**	0.888	0.889	**0.890**	**0.902**	0.900
income_spatial_OK	0.874	**0.907**	0.862	**0.881**	0.876	**0.884**	0.887	**0.891**	0.891	**0.895**
income_spatial_PR	0.881	**0.887**	0.904	**0.925**	0.908	**0.933**	0.900	**0.911**	0.895	**0.911**
income_spatial_TX	**0.917**	0.904	0.894	**0.905**	0.899	**0.904**	0.890	**0.894**	0.885	**0.892**
income_spatial_WA	0.908	**0.918**	0.898	**0.914**	0.904	**0.920**	0.894	**0.906**	0.901	**0.906**
income_spatial_WV	0.881	**0.889**	**0.923**	0.917	0.912	**0.913**	**0.914**	0.913	0.905	**0.910**
income_temporal_CA14	0.939	**0.949**	0.911	**0.919**	0.920	**0.922**	0.915	**0.916**	0.904	**0.908**
mobility_spatial_FL	0.862	**0.894**	0.891	**0.896**	0.893	**0.896**	0.897	**0.897**	0.901	**0.903**
mobility_spatial_MS	0.888	**0.947**	0.884	**0.914**	0.907	**0.921**	0.900	**0.923**	0.892	**0.915**
mobility_spatial_NY	0.854	**0.938**	0.873	**0.916**	0.879	**0.921**	0.881	**0.926**	0.893	**0.922**
mobility_spatial_OK	0.871	**0.916**	0.900	**0.940**	0.910	**0.950**	0.908	**0.954**	0.904	**0.949**
mobility_spatial_PR	**0.927**	0.920	0.893	**0.919**	0.885	**0.900**	0.886	**0.902**	0.897	**0.904**
mobility_spatial_TX	**0.927**	0.913	0.891	**0.915**	0.916	**0.920**	0.913	**0.920**	0.904	**0.917**
mobility_spatial_WA	0.861	**0.905**	0.898	**0.939**	0.920	**0.975**	0.918	**0.951**	0.916	**0.950**
mobility_spatial_WV	0.919	**0.968**	0.902	**0.969**	0.896	**0.953**	0.894	**0.954**	0.899	**0.943**
mobility_temporal_CA14	0.909	**0.928**	**0.915**	0.907	0.910	**0.911**	**0.906**	0.905	**0.908**	0.907
pubcov_spatial_FL	0.922	**0.935**	0.925	**0.925**	0.912	**0.922**	**0.911**	0.909	0.904	**0.907**
pubcov_spatial_MS	0.892	**0.936**	0.902	**0.917**	0.892	**0.901**	0.897	**0.909**	0.907	**0.913**
pubcov_spatial_NY	0.831	**0.882**	0.880	**0.914**	0.881	**0.909**	0.888	**0.900**	0.888	**0.893**
pubcov_spatial_OK	0.837	**0.862**	0.892	**0.907**	0.887	**0.905**	0.896	**0.898**	0.888	**0.894**
pubcov_spatial_PR	0.883	**0.926**	0.902	**0.953**	0.902	**0.957**	0.909	**0.969**	0.910	**0.964**
pubcov_spatial_TX	0.899	**0.899**	**0.904**	0.904	0.904	**0.908**	0.898	**0.907**	0.901	**0.905**
pubcov_spatial_WA	**0.925**	0.910	**0.927**	0.923	**0.921**	0.915	**0.904**	0.899	**0.906**	0.905
pubcov_spatial_WV	**0.919**	0.889	0.875	**0.897**	0.897	**0.904**	0.904	**0.908**	**0.904**	0.902
pubcov_temporal_CA14	0.842	**0.913**	0.880	**0.929**	0.897	**0.922**	0.901	**0.920**	0.911	**0.922**
traveltime_spatial_FL	**0.909**	0.898	0.916	**0.929**	0.912	**0.927**	0.911	**0.916**	**0.908**	0.908
traveltime_spatial_MS	0.939	**0.948**	0.902	**0.926**	0.916	**0.927**	0.901	**0.912**	0.897	**0.912**
traveltime_spatial_NY	**0.906**	0.889	0.917	**0.919**	**0.904**	0.898	0.909	**0.911**	0.899	**0.902**
traveltime_spatial_OK	0.903	**0.922**	0.906	**0.943**	0.886	**0.931**	0.892	**0.928**	0.910	**0.934**
traveltime_spatial_PR	0.916	**0.963**	0.906	**0.961**	0.889	**0.945**	0.884	**0.946**	0.894	**0.949**
traveltime_spatial_TX	0.886	**0.917**	0.876	**0.894**	0.903	**0.934**	0.909	**0.921**	0.899	**0.924**
traveltime_spatial_WA	0.897	**0.943**	0.905	**0.921**	0.923	**0.936**	0.913	**0.917**	0.910	**0.918**
traveltime_spatial_WV	**0.908**	0.865	0.887	**0.895**	**0.887**	0.882	0.899	**0.902**	0.888	**0.903**
traveltime_temporal_CA14	**0.909**	0.908	**0.912**	0.902	0.882	**0.894**	0.899	**0.910**	0.911	**0.922**
wine_r2w	0.937	**0.967**	0.907	**0.987**	0.903	**0.983**	0.901	**0.977**	0.904	**0.978**
mean	0.898	0.915	0.899	0.918	0.902	0.918	0.901	0.915	0.902	0.915
10th pctile	0.856	0.888	0.874	0.895	0.884	0.894	0.888	0.896	0.890	0.898

Table 32: Per-setting coverage, naive vs SASC, Logistic reg. (5 seeds, nominal 0.90). Bold marks the larger of the two.

setting	k=20		k=50		k=100		k=200		k=500	
	nv	SASC	nv	SASC	nv	SASC	nv	SASC	nv	SASC
employment_spatial_FL	0.909	**0.933**	0.904	**0.916**	0.903	**0.923**	0.909	**0.933**	0.902	**0.924**
employment_spatial_MS	0.909	**0.926**	0.906	**0.945**	0.907	**0.929**	0.911	**0.917**	0.897	**0.914**
employment_spatial_NY	0.824	**0.902**	0.866	**0.899**	0.866	**0.906**	0.894	**0.916**	0.895	**0.910**
employment_spatial_OK	0.884	**0.913**	0.912	**0.922**	0.921	**0.930**	0.923	**0.925**	0.908	**0.914**
employment_spatial_PR	0.895	**0.925**	0.884	**0.911**	0.883	**0.911**	0.892	**0.912**	0.904	**0.909**
employment_spatial_TX	0.928	**0.934**	**0.938**	0.916	**0.921**	0.921	0.918	**0.918**	0.907	**0.910**
employment_spatial_WA	0.852	**0.879**	0.856	**0.874**	0.896	**0.900**	0.908	**0.908**	0.909	**0.910**
employment_spatial_WV	0.938	**0.955**	0.904	**0.932**	0.914	**0.928**	0.912	**0.928**	0.906	**0.925**
employment_temporal_CA14	**0.926**	0.913	**0.935**	0.926	**0.925**	0.924	0.912	**0.916**	0.893	**0.895**
income_spatial_FL	**0.918**	0.918	0.896	**0.912**	0.911	**0.913**	0.901	**0.908**	0.903	**0.910**
income_spatial_MS	**0.927**	0.923	0.921	**0.925**	0.912	**0.921**	0.914	**0.928**	0.907	**0.917**
income_spatial_NY	0.851	**0.910**	0.854	**0.896**	0.877	**0.901**	0.876	**0.901**	0.891	**0.905**
income_spatial_OK	0.831	**0.915**	0.877	**0.915**	0.877	**0.918**	0.886	**0.905**	0.905	**0.910**
income_spatial_PR	0.869	**0.882**	**0.909**	0.891	0.913	**0.926**	0.899	**0.910**	0.890	**0.909**
income_spatial_TX	0.880	**0.918**	0.906	**0.907**	**0.920**	0.917	**0.907**	0.904	0.901	**0.905**
income_spatial_WA	**0.938**	0.923	0.917	**0.924**	0.906	**0.924**	0.904	**0.922**	0.907	**0.916**
income_spatial_WV	0.904	**0.934**	0.903	**0.925**	0.897	**0.924**	0.897	**0.913**	0.903	**0.917**
income_temporal_CA14	0.895	**0.918**	0.872	**0.909**	0.892	**0.906**	0.889	**0.903**	0.885	**0.902**
mobility_spatial_FL	0.840	**0.894**	0.892	**0.892**	0.883	**0.897**	0.894	**0.895**	**0.896**	0.896
mobility_spatial_MS	0.897	**0.913**	0.902	**0.916**	**0.911**	0.908	0.894	**0.896**	0.900	**0.904**
mobility_spatial_NY	0.887	**0.900**	0.874	**0.911**	0.877	**0.917**	0.890	**0.918**	0.886	**0.918**
mobility_spatial_OK	**0.881**	0.878	0.888	**0.899**	0.905	**0.913**	**0.913**	0.912	**0.910**	0.909
mobility_spatial_PR	0.941	**0.949**	0.894	**0.944**	0.887	**0.945**	0.903	**0.944**	0.898	**0.944**
mobility_spatial_TX	0.930	**0.957**	**0.927**	0.914	**0.911**	0.908	**0.913**	0.909	0.899	0.899
mobility_spatial_WA	0.889	**0.923**	**0.937**	0.932	0.930	**0.954**	0.921	**0.926**	0.916	**0.921**
mobility_spatial_WV	0.914	**0.935**	0.906	**0.924**	0.899	**0.905**	0.895	**0.899**	0.902	**0.902**
mobility_temporal_CA14	**0.924**	0.911	0.923	**0.927**	**0.915**	0.914	**0.908**	0.907	**0.904**	0.903

setting	k=20		k=50		k=100		k=200		k=500	
	nv	SASC	nv	SASC	nv	SASC	nv	SASC	nv	SASC
pubcov_spatial_FL	**0.933**	0.921	0.902	**0.930**	0.883	**0.917**	0.880	**0.906**	0.888	**0.904**
pubcov_spatial_MS	**0.930**	0.925	0.895	**0.908**	0.877	**0.916**	0.879	**0.909**	0.888	**0.908**
pubcov_spatial_NY	**0.893**	0.854	**0.894**	0.878	**0.878**	0.877	**0.887**	0.885	0.891	**0.892**
pubcov_spatial_OK	0.844	**0.896**	0.871	**0.900**	0.886	**0.907**	0.890	**0.900**	0.901	**0.908**
pubcov_spatial_PR	**0.932**	0.922	0.895	**0.962**	0.887	**0.957**	0.897	**0.958**	0.897	**0.962**
pubcov_spatial_TX	0.856	**0.915**	0.873	**0.909**	0.864	**0.917**	0.881	**0.914**	0.897	**0.915**
pubcov_spatial_WA	**0.916**	0.890	0.883	**0.911**	0.893	**0.899**	**0.898**	0.887	**0.902**	0.901
pubcov_spatial_WV	0.864	**0.872**	0.884	**0.887**	0.886	**0.892**	**0.904**	0.899	**0.901**	0.899
pubcov_temporal_CA14	0.874	**0.907**	0.904	**0.914**	**0.914**	0.913	0.906	**0.917**	0.910	**0.915**
traveltime_spatial_FL	**0.928**	0.917	0.928	**0.929**	0.925	**0.929**	0.911	**0.921**	0.907	**0.917**
traveltime_spatial_MS	0.883	**0.947**	0.900	**0.949**	0.902	**0.938**	0.913	**0.944**	0.899	**0.935**
traveltime_spatial_NY	**0.912**	0.909	**0.911**	0.907	0.903	**0.907**	**0.896**	0.893	**0.893**	0.892
traveltime_spatial_OK	0.900	**0.919**	0.928	**0.936**	0.914	**0.933**	0.898	**0.926**	0.900	**0.927**
traveltime_spatial_PR	0.879	**0.899**	0.910	**0.922**	0.892	**0.915**	0.889	**0.910**	0.889	**0.917**
traveltime_spatial_TX	0.906	**0.920**	0.853	**0.906**	0.901	**0.917**	0.914	**0.931**	0.913	**0.925**
traveltime_spatial_WA	**0.928**	0.926	0.920	**0.944**	0.918	**0.934**	0.908	**0.920**	0.902	**0.917**
traveltime_spatial_WV	0.902	**0.906**	0.890	**0.924**	0.901	**0.922**	0.901	**0.919**	0.897	**0.919**
traveltime_temporal_CA14	0.875	**0.882**	**0.898**	0.895	0.892	**0.899**	0.895	**0.905**	0.891	**0.902**
wine_r2w	0.909	**0.929**	0.898	**0.912**	0.884	**0.911**	0.883	**0.905**	0.896	**0.910**
mean	0.897	0.914	0.899	0.916	0.899	0.917	0.900	0.914	0.900	0.912
10th pctile	0.851	0.882	0.872	0.894	0.877	0.899	0.885	0.898	0.890	0.899

N COVERAGE LEVEL: DOES THE STORY DEPEND ON $\alpha = 0.1$?

The main text fixes the nominal level at 90%. A conformal quantile at a different level probes a different part of the score distribution, and the anchor's bias is defined *at* the level being estimated (Def. 1). Thus there is no a priori reason for the conclusion to transfer. Table 33 and Figure 16 repeat the comparison at $\alpha \in \{0.05, 0.10, 0.20\}$.

The ordering survives at every level for $k \geq 50$, but the *magnitude* of the gain moves against the direction the asymptotic theory suggests. Asymptotically, a smaller α probes a sparser part of the tail. This inflates $v_n = \alpha(1-\alpha)/f_T(q_T)^2$ and so, by Eq. (25), leaves the anchor *more* to contribute. Empirically the gain shrinks as α falls. At k=50 SASC improves tail coverage over naive by $+0.019$ at $\alpha = 0.2$, $+0.013$ at $\alpha = 0.1$ and only $+0.007$ at $\alpha = 0.05$. Finite-sample discreteness overrides the asymptotics. The split conformal rank is $\lceil (k+1)(1-\alpha) \rceil$. At small α with small k that rank sits against the top of the sample. At $\alpha = 0.05, k = 20$ it is 20 out of 20, so naive *is* the sample maximum and over-covers by construction, leaving no deficit for the anchor to repair. This is the one regime in the study where SASC sits below naive in the tail (0.905 vs 0.921 at α=0.05, k=20, both models). Any interior blend of an already-maximal threshold with a finite anchor can only lower it. The effect is a property of the budget, not of the method, and it disappears by $k = 50$. It also refines the practical guidance: SASC's benefit is largest when the label budget can resolve the requested level, that is when $k(1-\alpha)$ is well inside the sample. The conservative variant follows Table 3 exactly. No order statistic attains the guarantee at $\alpha = 0.05, k = 20$ (hence full sets), the required rank is $50/50$ at $k = 50$, and its width premium grows as the level tightens.

O NONCONFORMITY SCORE: LAC, APS, RAPS

The main text uses LAC, which gives the smallest sets but is known to be poor at class-conditional coverage. APS (Romano et al., 2020) and RAPS (Angelopoulos et al., 2021) trade size for adaptivity. The ACS and Wine tasks are binary, and APS and RAPS largely degenerate there. We therefore run this ablation on the induced-shift OpenML family, which contains multiclass problems (up to 7 classes; see Table 7). Table 35 and Figure 17 report the comparison.

The result is a real limitation and also the clearest confirmation of the theory in the paper. Under LAC, SASC improves tail coverage as it does on the concept-shift suite (0.868 vs naive's 0.855 at k=20 for TabICL; 0.878 vs 0.858 for TabPFN). Under APS it is a *wash*, within ± 0.01 of naive at every budget. Under RAPS at k=20 it is clearly *worse* (0.784 vs 0.841, TabICL): the anchor stops paying when the score changes.

Prop. 4 says when this must happen. The anchor's variance advantage is $\kappa = \alpha(1-\alpha)/f_T(q_T)^2 - 1/(4f_T(m_T)^2)$. It is positive only when the score density at the $(1-\alpha)$ quantile is small relative to

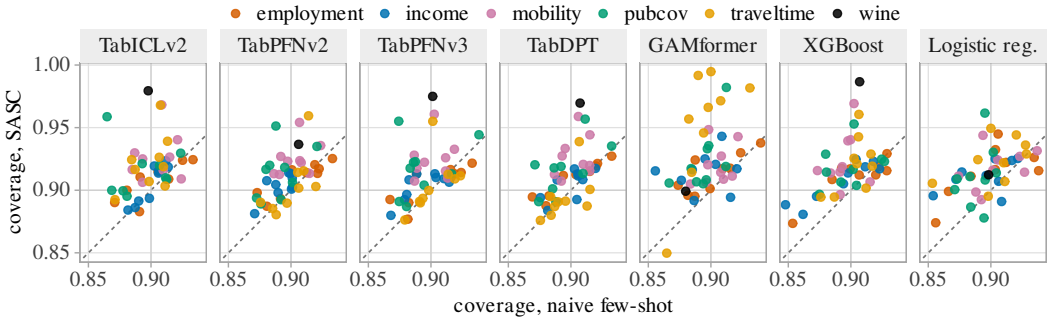


Figure 14: **Paired per-setting comparison at k=50, all five base models.** Each point is one of the 46 concept-shift settings; points above the diagonal are settings where SASC covers better than naive few-shot conformal. The improvement is concentrated below and to the left, among the settings that were undercovering. Points already at or above nominal move very little, which is the self-limiting behavior the cross-validated weight is designed to produce. Color marks the ACS task.

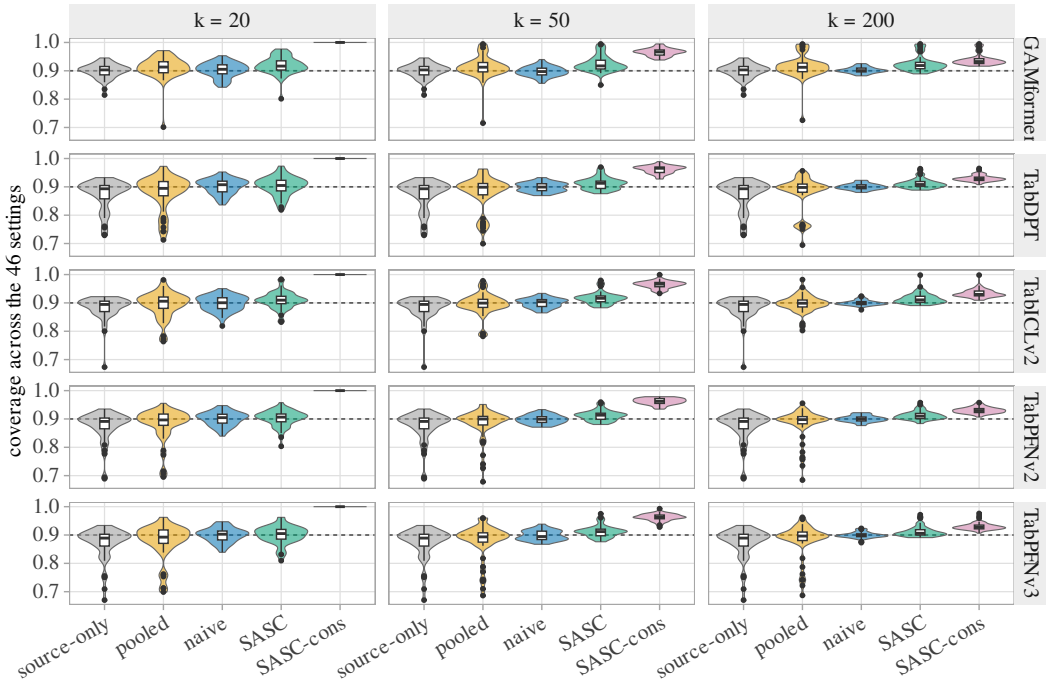


Figure 15: Per-setting coverage for five calibration rules at three budgets (violins with box plots). The 10th percentile, our headline statistic, is the bottom of each distribution.

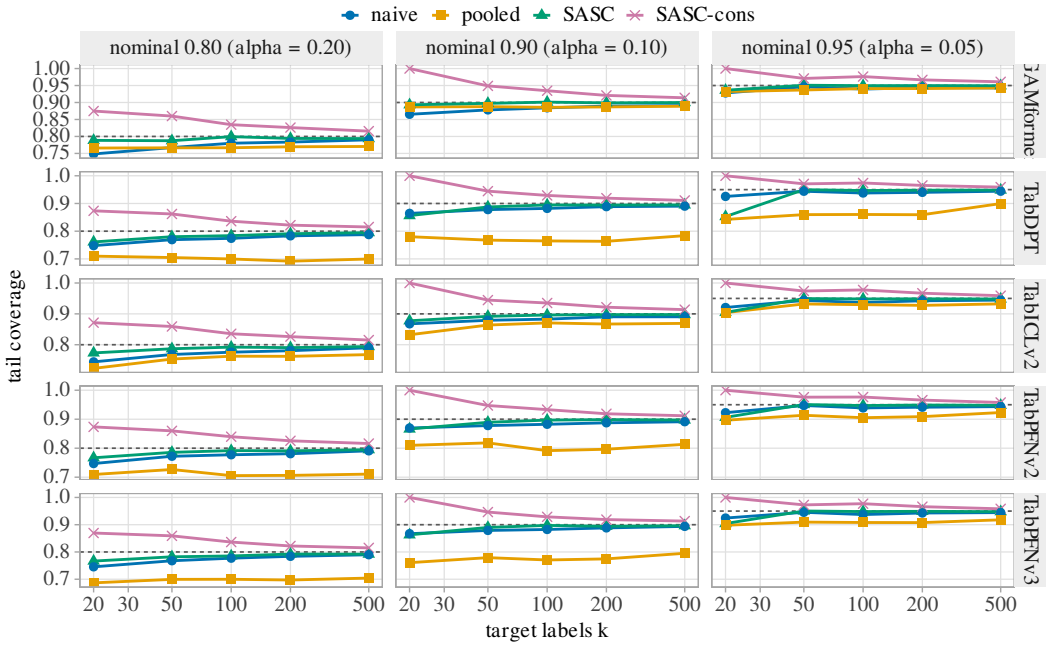


Figure 16: **The comparison at three nominal levels.** Tail coverage against label budget for $\alpha \in \{0.05, 0.1, 0.2\}$ (columns) and both TFMs (rows); the dashed line in each panel is that panel's nominal level. The ordering is preserved at every level for $k \geq 50$, but the size of SASC's advantage grows as the level loosens. At $\alpha = 0.05, k = 20$ the split-conformal rank $\lceil (k+1)(1-\alpha) \rceil$ equals the sample size, so naive *is* the sample maximum and over-covers by construction, and SASC sits below it. That single crossing is a property of the budget, not of the method.

the density at the median, that is when $f_T(m_T)/f_T(q_T) > 1/(2\sqrt{\alpha(1-\alpha)}) = 1.67$ at $\alpha = 0.1$. LAC scores are strongly right-skewed with a sparse upper tail (most points are easy, few are hard). We measure $f_T(m_T)/f_T(q_T) \approx 2.5$ (Table 47), so $\kappa > 0$. APS and RAPS are cumulative-mass scores. They pile probability toward 1, so their upper region is *dense*, the ratio collapses, and κ falls to near zero or below. Table 34 reports the measured ratio and κ per score and confirms this.

We take two things from this. First, a scope statement. SASC is a method for scores with a sparse upper tail. That includes the standard LAC and absolute-residual choices, and, per §V, normalized residuals and CQR. It does not include the cumulative-mass family; with APS/RAPS a practitioner should use naive few-shot conformal. SASC's CV gate does at least prevent a disaster, since it mostly reverts to naive rather than committing to a bad anchor. The exception is the k=20 RAPS case, where the gate itself is too noisy to protect. Second, a methodological point. κ is computable from unlabeled source scores and a handful of target labels, so a practitioner can check whether the anchor will help for a given score before deploying. The pooled baseline, by contrast, collapses under APS/RAPS (tail coverage 0.47–0.62), because pooling a location-shifted copy of a dense-tailed score distribution corrupts the quantile outright.

P Inside SASC: folds, grid, anchor statistic, and the fixed-weight rule

SASC has three internal choices: the number of cross-validation folds F, the candidate weight grid $\mathcal{W}$, and the statistic defining the anchor's location. Its predecessor is the fixed-weight rule $w_k = k/(k + K_0)$, which the adaptive rule replaced. Tables 36, 37, 38 and 39 and Figure 18 vary each in turn, holding everything else at the default. Figure 19 shows what the resulting gate selects.

Folds ($F \in \{3, 5, 10\}$). The method is nearly insensitive to F: across the budget range the tail coverage of $F \in \{3, 5, 10\}$ spans at most 0.007 (at k=20: 0.883/0.878/0.876) and set size spans 0.6%. What F does change is how much anchor the gate admits. The mean selected weight at k=50 falls from $\bar{w} = 0.38$ (F=3) to 0.25 (F=10), because more folds leave fewer held-out points per fold

Table 33: Coverage-level sweep: tail (10th-percentile) coverage across the 46 concept-shift settings at three nominal levels (5 seeds). "size" is the mean set size at k=50. Nominal coverage is $1-\alpha$, shown in the first column.

α	method	k=20	k=50	k=100	k=200	k=500	size$_{k=50}$
TabICLv2							
0.95	naive	0.921	0.943	0.937	0.942	0.945	1.65
	pooled	0.904	0.932	0.930	0.928	0.932	1.60
	SASC	0.905	0.950	0.948	0.949	0.948	1.67
	SASC-cons	1.000	0.974	0.978	0.967	0.959	1.79
0.90	naive	0.868	0.879	0.883	0.890	0.891	1.42
	pooled	0.832	0.864	0.871	0.867	0.869	1.41
	SASC	0.878	0.892	0.896	0.897	0.898	1.46
	SASC-cons	1.000	0.945	0.935	0.922	0.914	1.66
0.80	naive	0.744	0.768	0.776	0.781	0.790	1.15
	pooled	0.723	0.754	0.763	0.762	0.768	1.15
	SASC	0.773	0.787	0.792	0.791	0.794	1.20
	SASC-cons	0.871	0.859	0.836	0.826	0.815	1.37
TabPFNv2							
0.95	naive	0.922	0.947	0.939	0.941	0.943	1.64
	pooled	0.896	0.914	0.905	0.909	0.923	1.57
	SASC	0.906	0.951	0.947	0.949	0.948	1.65
	SASC-cons	1.000	0.976	0.977	0.966	0.958	1.78
0.90	naive	0.870	0.878	0.882	0.888	0.891	1.40
	pooled	0.810	0.818	0.792	0.796	0.813	1.36
	SASC	0.866	0.889	0.897	0.899	0.897	1.44
	SASC-cons	1.000	0.947	0.933	0.919	0.912	1.65
0.80	naive	0.747	0.772	0.777	0.781	0.791	1.15
	pooled	0.709	0.727	0.705	0.706	0.711	1.11
	SASC	0.767	0.786	0.792	0.791	0.795	1.18
	SASC-cons	0.874	0.860	0.840	0.826	0.816	1.35

and a noisier per-fold estimate. The coverage consequences are within noise. We keep $F = 5$, and the choice does no work.

Weight grid. The binary grid $\{0, 1\}$ (use the anchor or do not) is not worse. It gives the best tail coverage at small k (0.890 vs 0.878 at k=20). It is also the most conservative choice, with the largest sets (1.478 vs 1.464 at k=50) and the highest mean coverage (0.921 vs 0.917), so it buys tail reliability with width rather than for free. Refining the grid from 8 to 21 points moves nothing ($\leq$ 0.007 coverage). The grid thus selects a point on a coverage/width trade rather than a better method, and SASC does not depend on a finely tuned hyperparameter that must be re-selected per deployment; the default eight-point grid is a middle setting.

Anchor statistic. The median-gap anchor gives the best tail coverage of the five (0.878/0.892/0.896 at k=20/50/100). The two-parameter location–scale anchor is *strictly dominated*, with worse coverage (0.847 at k=20) at *larger* sets (1.473 vs 1.464). The mean-gap, trimmed-mean and 75th-percentile anchors sit in between, trading a little coverage for slightly tighter sets. The failure of the richer anchor mirrors the negative results of §4.3. Estimating a scale factor from k target points adds variance, and it buys a correction to a component of the shift that §U measures to be small. The extra parameter is therefore a net loss, and as before, more elaborate is not automatically better; the pattern repeats inside our own method.

The fixed-weight rule. Table 39 varies $K_0 \in \{10, 20, 40, 80\}$ in $w_k = k/(k + K_0)$. The adaptive rule beats every value at every budget. Which constant wins is informative: it is $K_0 = 10$, the setting that uses the anchor *least* ($\bar{w} = 0.83$ at k=50). Its 0.882 tail coverage at size 1.416 for k=50 is barely distinguishable from naive's 0.879 at 1.420. In other words, the best fixed schedule is the one that nearly switches the anchor off. The adaptive rule uses the anchor far more ($\bar{w} = 0.30$) and still does better (0.892 at 1.464), because it uses it *selectively*, heavily where the anchor is good and not at

Table 34: Why the anchor helps for some scores and not others. $\kappa > 0$ (equivalently $f_T(m_T)/f_T(q_T) > 1.67$ at α=0.1) is Prop. 4's condition for the anchor to have any variance advantage. Measured on the induced-shift OpenML family.

model	score	$f_T(m_T)/f_T(q_T)$	κ (median)	frac. $\kappa > 0$	anchor helps?
TabICLv2	LAC	7.32	+0.0548	0.84	yes
	APS	1.01	-0.0072	0.06	no
	RAPS	1.01	-0.0005	0.09	no
TabPFNv2	LAC	5.89	+0.0576	0.81	yes
	APS	1.00	-0.0352	0.00	no
	RAPS	1.00	-0.0401	0.00	no
TabPFNv3	LAC	5.40	+0.0563	0.82	yes
	APS	1.00	-0.0004	0.09	no
	RAPS	1.00	-0.0001	0.11	no
TabDPT	LAC	7.92	+0.0578	0.83	yes
	APS	1.01	-0.0157	0.06	no
	RAPS	1.01	-0.0052	0.06	no
GAMformer	LAC	5.56	+0.0880	0.83	yes
	APS	1.00	-0.0136	0.03	no
	RAPS	1.02	-0.0069	0.11	no
XGBoost	LAC	8.77	+0.1183	0.80	yes
	APS	1.02	-0.0027	0.00	no
	RAPS	1.16	-0.0001	0.14	no
Logistic reg.	LAC	4.84	+0.0429	0.70	yes
	APS	0.95	-0.0272	0.00	no
	RAPS	1.22	-0.0053	0.20	no

all where it is not. That is the behavior Prop. 4 calls for: the risk-optimal weight depends on the unknown anchor bias b and not on k alone, so no better constant can repair the fixed schedule. The extra reliability is not free: SASC pays about 3% in set size over the fixed rule and naive.

Q THE CONSERVATIVE FORM: CHOOSING δ

Table 40 and Figure 20 sweep the training-conditional confidence $\delta \in \{0.01, 0.05, 0.10, 0.20\}$; the empirical behavior tracks Eq. (20). Tightening δ from 0.2 to 0.01 moves the required order statistic by only five ranks at k=200 (Table 3). The width penalty is thus mild: mean set size rises from 1.50 to 1.58, about 5%, for a much stronger certificate. At k=50 the same change costs 12% ($1.58 \to 1.77$). The cliff is at k=20, where only $\delta = 0.2$ admits any valid rank; every stricter δ returns the full label set (mean size 2.00 on these binary tasks), so the certificate says nothing. In practice, at $k \lesssim 20$, δ is not the knob to turn; k is. From $k \approx 100$ upward a strict $\delta = 0.01$ costs a few percent of width, and we prefer it whenever a certificate is the point of the conservative form.

R HOW MUCH SOURCE CALIBRATION DATA DOES THE ANCHOR NEED?

The anchor assumes $n_s \gg k$: the source quantile is close to noiseless relative to the target statistics. §D.3 quantifies the neglected term as $O(1/n_s)$ against the anchor's own $O(1/k)$. Table 41 and Figure 21 test this directly: we subsample the source calibration set to $n_s \in \{250, 500, 1000, 2000, 3000\}$ with the predictor fixed. This isolates calibration-set size from predictor quality, which the 50k-scale experiment of §G deliberately conflates.

Shrinking the source calibration set twelvefold, from 3000 to 250, costs SASC almost nothing. At k=50 its tail coverage falls only from 0.892 to 0.890, still well above naive's 0.879, and the advantage holds at every n_s for $k \geq 50$. The anchor runs out of information only at the smallest budget: at k=20 with n_s=250, SASC exactly ties naive (0.868), after gaining $+0.010$ at n_s=3000. At that boundary the $O(1/n_s)$ source term of §D.3 becomes comparable to the anchor's own $O(1/k)$ variance, which at $n_s/k \approx 12$ is about where we expect it. In practice a few hundred source calibration points are enough to anchor with, and when they are not, the CV gate reverts to naive rather than degrading.

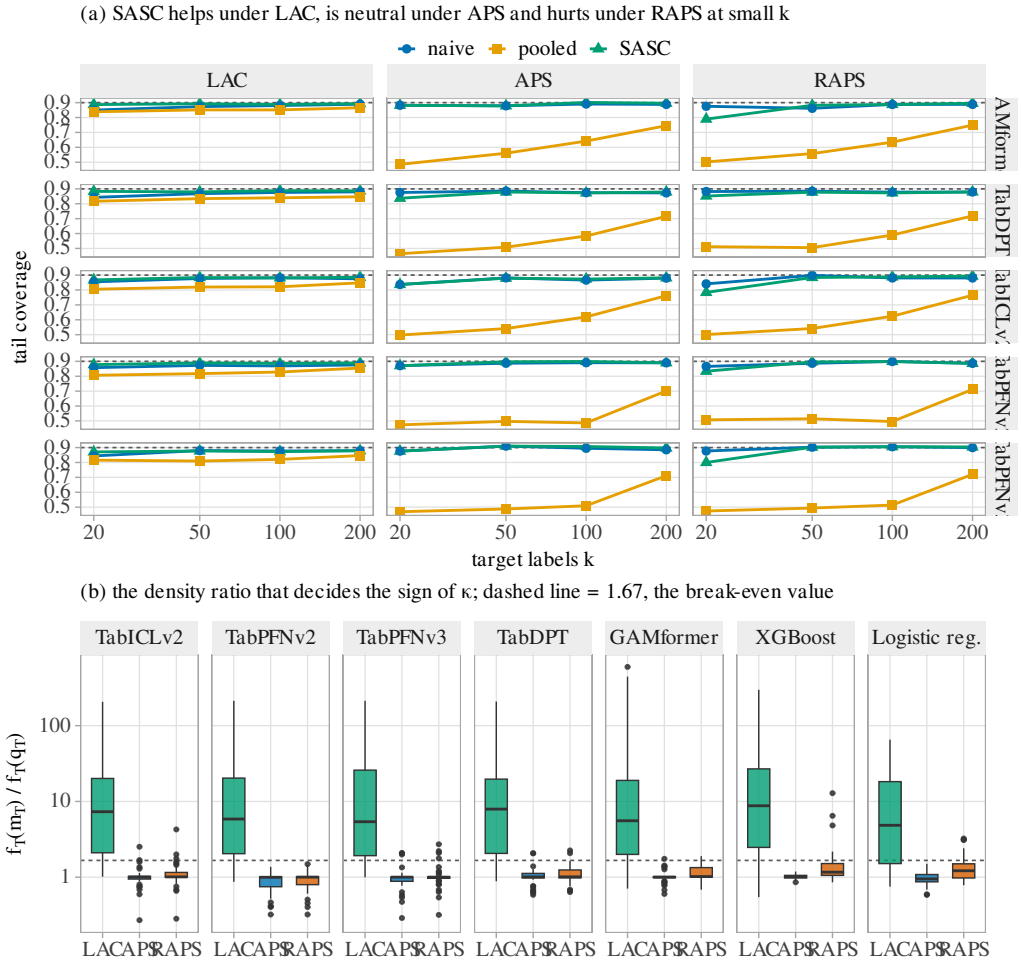


Figure 17: *(a)* Tail coverage on the multiclass induced-shift family for the three classification scores. *(b)* The ratio of the target score density at the median to that at the $(1-\alpha)$ quantile, by score; Prop. 4 requires it to exceed 1.67 (dashed) for the anchor to help. LAC is above the line, APS and RAPS at or below it.

S A SECOND REAL-SHIFT FAMILY FROM PUBLIC UCI TABLES

All real shift in the main suite is census-derived, apart from Wine. To test the paper's claims on unrelated data, we build eleven further settings from six public tables (Vanschoren et al., 2014). Each has a documented domain split of the kind used by TableShift (Gardner et al., 2023), and we remove the split feature from the predictors. *Diabetes readmission* (OpenML 4541; 101k hospital encounters): source is patients admitted by physician, clinic or HMO referral; targets are emergency-room admissions and transfers from another facility; the label is any readmission. *Forest cover type* (OpenML 1596): source is wilderness area 1; targets are areas 2, 3 and 4; the four cover types present in the source define the label space. *Electricity* (OpenML 151): source is the first 35% of the recording period; targets are the middle 30% and the last 35%. *Bank marketing* (OpenML 1461): source is contacts made in May to August; target is the remaining months. *Adult* (OpenML 1590): source is US-born respondents; target is the foreign-born. *Gas sensor array drift* (UCI, 13,910 measurements of six gases over 36 months in ten batches): source is batches 1–3; targets are batches 4–6 and 7–10, a sensor drift that is neither person- nor event-level. Every setting uses the protocol of §L ($n_{\text{tr}} = n_s = 3000$, 500-point target pool, 2500-point test set, five seeds, five base models).

These shifts are more severe than ACS. Source-only calibration attains a tail coverage of 0.67–0.69 for all five TFMs on the nine tabular settings. On the two sensor-drift settings it covers almost nothing;

Table 35: Nonconformity-score ablation on the induced-shift OpenML family (multiclass; 5 seeds). Tail (10th-percentile) coverage across datasets, and mean set size at k=50. Nominal coverage is 0.90.

score	method	k=20	k=50	k=100	k=200	size$_{k=50}$
TabICLv2						
LAC	naive	0.855	0.877	0.882	0.875	1.36
	pooled	0.805	0.820	0.822	0.848	1.37
	SASC	0.868	0.884	0.881	0.886	1.48
	SASC-cons	1.000	0.956	0.921	0.908	1.62
APS	naive	0.837	0.881	0.866	0.880	2.89
	pooled	0.497	0.541	0.619	0.762	2.26
	SASC	0.837	0.879	0.874	0.882	3.09
	SASC-cons	1.000	0.954	0.948	0.925	3.47
RAPS	naive	0.841	0.898	0.881	0.881	2.38
	pooled	0.501	0.541	0.624	0.765	1.88
	SASC	0.784	0.884	0.888	0.893	2.40
	SASC-cons	1.000	0.947	0.943	0.924	2.67
TabPFNv2						
LAC	naive	0.858	0.873	0.869	0.876	1.37
	pooled	0.806	0.818	0.828	0.855	1.44
	SASC	0.878	0.889	0.886	0.889	1.52
	SASC-cons	1.000	0.953	0.929	0.919	1.66
APS	naive	0.871	0.887	0.891	0.890	2.96
	pooled	0.473	0.497	0.486	0.701	2.37
	SASC	0.871	0.897	0.899	0.890	3.19
	SASC-cons	1.000	0.947	0.930	0.909	3.52
RAPS	naive	0.866	0.887	0.899	0.886	2.45
	pooled	0.507	0.514	0.496	0.711	1.90
	SASC	0.834	0.896	0.899	0.886	2.45
	SASC-cons	1.000	0.952	0.937	0.914	2.69

Table 36: Cross-validation folds F. Tail coverage across the 46 concept-shift settings; $\bar{w}$ is the mean selected shrinkage weight at k=50 (w=1 is pure naive, w=0 is pure anchor).

model	CV folds F	k=20	k=50	k=100	k=200	k=500	$\bar{w}_{k=50}$
TabICLv2	3	0.883	0.894	0.897	0.898	0.898	0.38
	5	0.878	0.892	0.896	0.897	0.898	0.30
	10	0.876	0.890	0.896	0.899	0.898	0.25
TabPFNv2	3	0.874	0.890	0.894	0.898	0.897	0.39
	5	0.866	0.889	0.897	0.899	0.897	0.33
	10	0.866	0.888	0.898	0.898	0.897	0.29

the source model's scores no longer describe the target. The anchor bias is an order of magnitude larger (median $|b|$ of 0.24 and 0.29 against 0.024 on ACS, Table 44), and $\kappa > 0$ in 75% and 71% of settings. With eleven settings the 10th percentile is the second-smallest value, so the sensor-drift pair sets the family's tail statistics. By Eq. (25) the predicted crossover $k^\star$ is thus small: the criterion forecasts little gain from source anchoring beyond what the target labels give. Tables 42 and 43 and Figure 22 report the outcome. Naive few-shot is reliable from k=50 on, and NexCP-gated matches it. SASC matches it for TabICL, XGBoost and logistic regression at $k \geq 50$ at a 5–10% size premium and trails slightly for TabPFN. At k=20 both gated rules sit at or below naive, since the held-out check has too few points on shifts this severe. The label-free and covariate-weighted rules collapse: pooled-DR and WCC-CSPD attain 0.58–0.63, below even source-only calibration on several settings, because covariate reweighting upweights the wrong points when the conditional has moved. The one setting where every rule exceeds nominal, cover type area 4, has a single class in the target, so any threshold that admits that class covers.

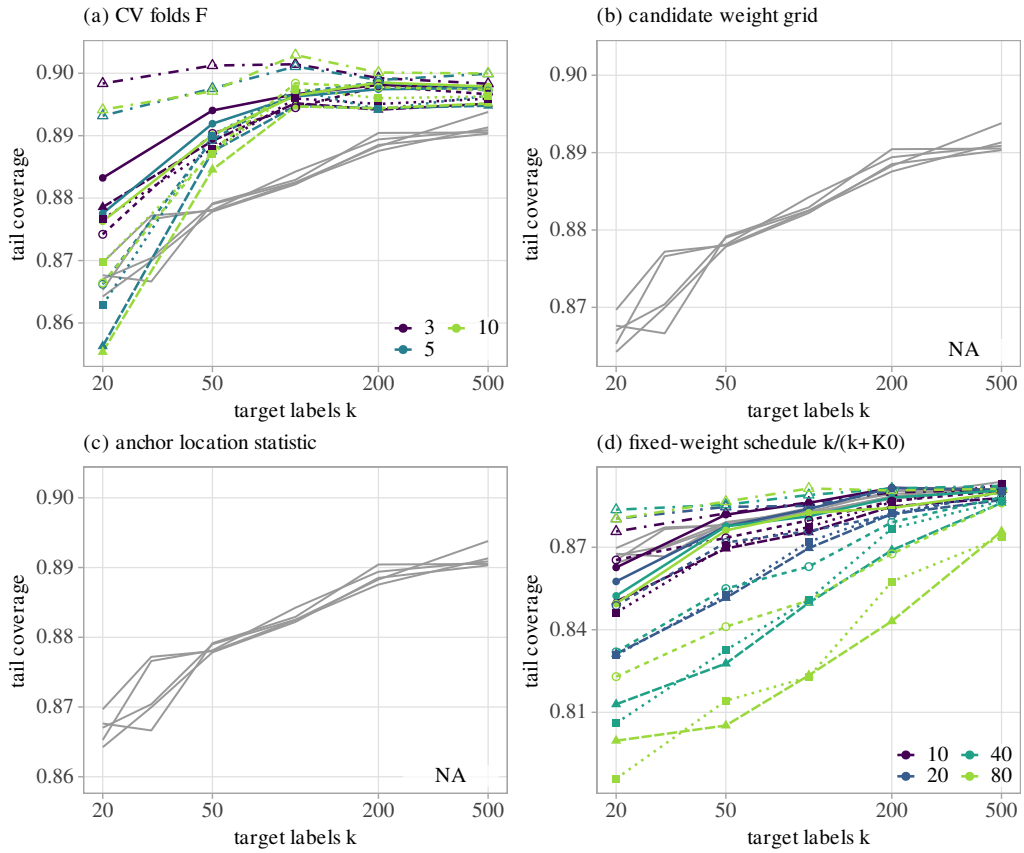


Figure 18: Tail coverage against budget for each internal choice of SASC, TabICLv2 (solid) and TabPFNv2 (dashed); gray is naive few-shot. *(a)* Number of folds. *(b)* Weight grid. *(c)* Anchor statistic. *(d)* Fixed schedules $w_k = k/(k + K_0)$ against the gate.

Table 37: Candidate weight grid $\mathcal{W}$. "binary" is $\{0, 1\}$, "coarse" five points, "default" the eight-point grid of Algorithm 1, "fine" 21 points.

model	weight grid $\mathcal{W}$	k=20	k=50	k=100	k=200	k=500	$\bar{w}_{k=50}$
TabICLv2	binary	0.890	0.895	0.899	0.899	0.898	0.40
	coarse	0.880	0.894	0.897	0.898	0.898	0.32
	default	0.878	0.892	0.896	0.897	0.898	0.30
	fine	0.871	0.891	0.896	0.897	0.897	0.29
TabPFNv2	binary	0.887	0.894	0.900	0.899	0.898	0.41
	coarse	0.871	0.889	0.898	0.899	0.897	0.34
	default	0.866	0.889	0.897	0.899	0.897	0.33
	fine	0.863	0.887	0.896	0.898	0.897	0.31

T IS THIS A FOUNDATION-MODEL PHENOMENON? A THIRD TFM AND TWO NON-TFM BASE MODELS

Two objections point in opposite directions. The first is that the coverage failure and the fix might be artifacts of in-context learning: the support set comes from the source, and the amortized synthetic prior never anticipated a shifted conditional. The second is that TabICL and TabPFN are one lineage, both amortized over synthetic priors, so two of them are one observation about how tabular foundation models are pretrained.

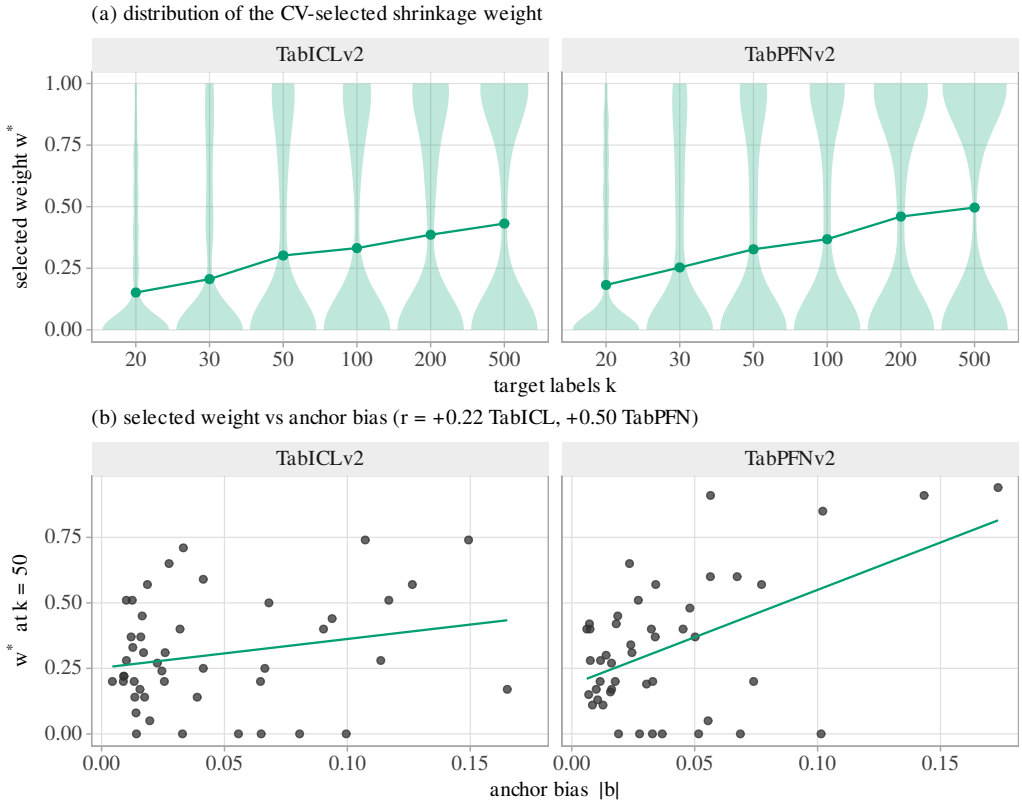


Figure 19: *(a)* Distribution of the selected weight $w^\star$ across settings and seeds, by budget; it is bimodal and its mean rises with k. *(b)* The selected weight against the anchor bias ($r = +0.22$ TabICLv2, $+0.50$ TabPFNv2).

Table 38: Anchor statistic. "median"/"mean"/"trim"/"q75" shift the source quantile by the corresponding location gap; "locscale" additionally matches inter-quartile ranges.

model	anchor statistic	k=20	k=50	k=100	k=200	k=500	$\bar{w}_{k=50}$
TabICLv2	median	0.878	0.892	0.896	0.897	0.898	0.30
	mean	0.869	0.885	0.890	0.897	0.895	0.28
	trim	0.869	0.888	0.893	0.897	0.896	0.27
	q75	0.865	0.889	0.892	0.897	0.897	0.32
	locscale	0.847	0.877	0.888	0.897	0.898	0.40
TabPFNv2	median	0.866	0.889	0.897	0.899	0.897	0.33
	mean	0.868	0.886	0.896	0.897	0.895	0.33
	trim	0.868	0.887	0.895	0.898	0.896	0.31
	q75	0.863	0.880	0.891	0.894	0.897	0.32
	locscale	0.844	0.876	0.887	0.897	0.897	0.36

We answer both with base models that sit on either side. Against the first objection we re-run the concept-shift suite with gradient-boosted trees (XGBoost (Chen & Guestrin, 2016), 300 trees) and ℓ_2-regularized logistic regression. Against the second we add three further TFMs: TabPFNv3 (Grinsztajn et al., 2026), the successor of TabPFNv2 in the same synthetic-prior lineage, which tests whether a newer model of the same kind changes anything; TabDPT (Ma et al., 2025), pretrained on real tables rather than synthetic priors and retrieving its context at prediction time, and GAMformer (Mueller et al., 2024), which emits the shape functions of a generalized additive model and is therefore structurally constrained. All seven base models are fit on the same source training rows and calibrated by identical code. Table 45 and Figure 23 report the result; Figure 24 places every calibration rule on the coverage/size frontier.

Table 39: The fixed-weight predecessor $w_k = k/(k + K_0)$. The adaptive rule beats every constant at every budget, and the best constant (K_0=10) is the one that uses the anchor least, landing on naive. The schedule has the wrong shape, because the risk-optimal weight depends on the anchor bias (Prop. 4), not on k alone.

model	fixed-weight K_0	k=20	k=50	k=100	k=200	k=500	$\bar{w}_{k=50}$
TabICLv2	10	0.863	0.882	0.886	0.892	0.891	0.83
	20	0.858	0.878	0.884	0.892	0.891	0.71
	40	0.852	0.878	0.881	0.888	0.891	0.56
	80	0.850	0.876	0.883	0.884	0.890	0.38
TabPFNv2	10	0.865	0.873	0.880	0.887	0.891	0.83
	20	0.849	0.872	0.876	0.882	0.890	0.71
	40	0.832	0.855	0.863	0.879	0.888	0.56
	80	0.823	0.841	0.851	0.868	0.886	0.38

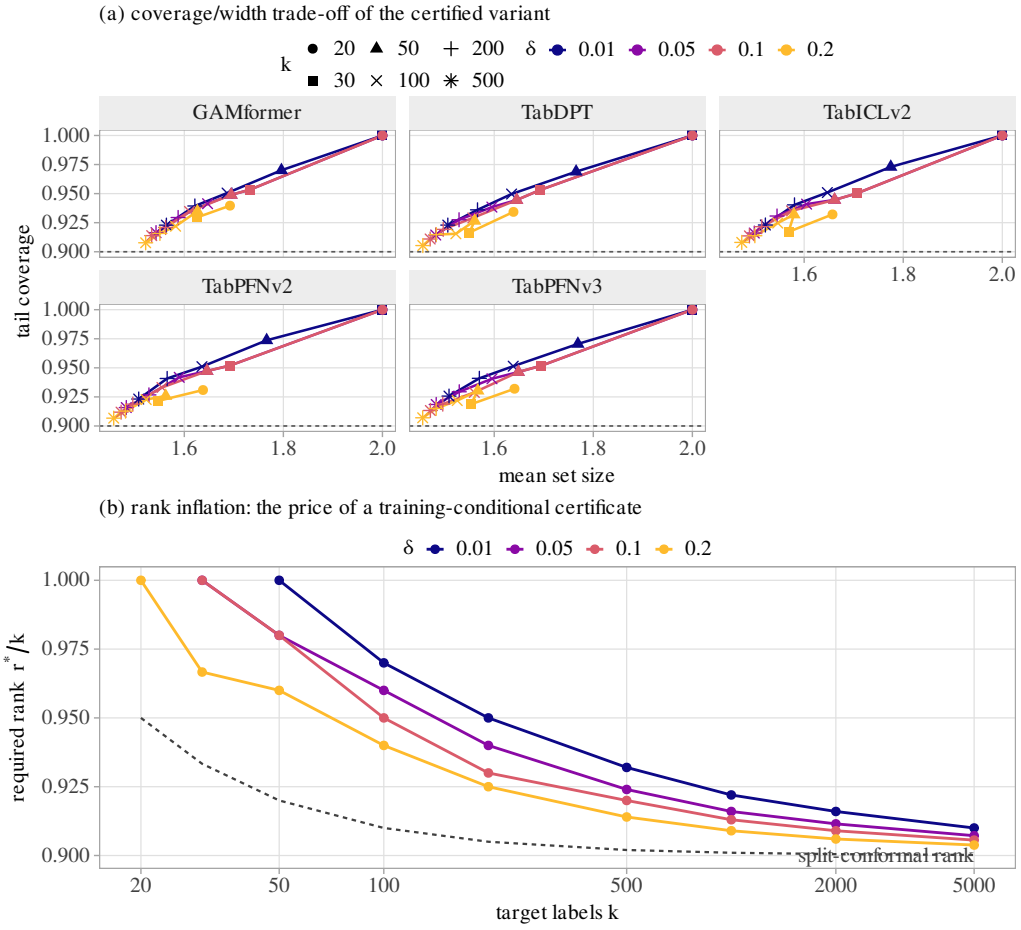


Figure 20: *(a)* Coverage against set size for the conservative form; each path runs over k at a fixed δ. *(b)* The required rank $r^\star/k$ from Prop. 3 against the split-conformal rank (dashed); at k=20 no rank attains $\delta \leq 0.1$.

Neither the failure nor the fix depends on the pretraining recipe, and a newer model of the same recipe changes nothing: TabPFNv3 loses coverage under the induced split (0.903 $\rightarrow$ 0.737), sits at a closest-to-nominal oracle gap of 0.026 on ACS like its predecessor, has the largest anchor bias of the five (median $|b| = 0.029$), and is hurt most by plain pooling (0.779). TabDPT loses coverage under the induced split exactly as the other two do, from 0.902 to 0.733 against 0.697 for TabICL and 0.775 for TabPFN (Table 11), and on the 46-setting suite its source-only calibration sits at 0.792,

Table 40: Training-conditional confidence δ for SASC-cons: tail coverage and mean set size across the 46 settings. Compare Table 3: entries at k=20 are degenerate (full sets) whenever no order statistic attains the guarantee. Nominal coverage is 0.90.

model	δ	metric	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	0.01	tail cov	1.000	1.000	0.973	0.951	0.940	0.923
		mean size	2.00	2.00	1.77	1.65	1.58	1.52
	0.05	tail cov	1.000	0.950	0.945	0.941	0.931	0.916
		mean size	2.00	1.71	1.66	1.60	1.55	1.50
	0.10	tail cov	1.000	0.950	0.945	0.935	0.922	0.914
		mean size	2.00	1.71	1.66	1.57	1.51	1.49
	0.20	tail cov	0.932	0.917	0.932	0.925	0.914	0.908
		mean size	1.66	1.57	1.58	1.55	1.50	1.47
TabPFNv2	0.01	tail cov	1.000	1.000	0.974	0.951	0.941	0.923
		mean size	2.00	2.00	1.77	1.64	1.57	1.51
	0.05	tail cov	1.000	0.952	0.947	0.942	0.927	0.916
		mean size	2.00	1.69	1.65	1.59	1.53	1.48
	0.10	tail cov	1.000	0.952	0.947	0.933	0.919	0.912
		mean size	2.00	1.69	1.65	1.55	1.50	1.47
	0.20	tail cov	0.931	0.922	0.926	0.923	0.913	0.907
		mean size	1.64	1.55	1.56	1.52	1.48	1.46

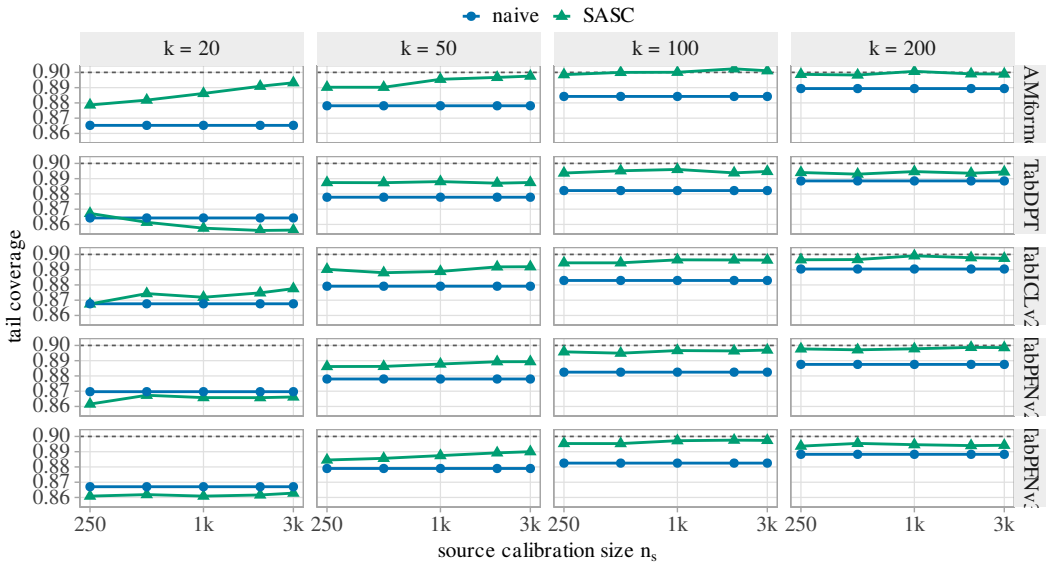


Figure 21: **The anchor needs little source data.** Tail coverage against source calibration size, with the predictor fixed so that only the calibration sample varies. Naive few-shot is flat by construction (it never looks at the source). SASC degrades only at the smallest budget: at $k \geq 50$ its advantage is intact even with the source set shrunk twelvefold to 250 points; the two curves meet only at k=20, where $n_s/k \approx 12$ and the $O(1/n_s)$ source term of §D.3 becomes comparable to the anchor's own $O(1/k)$ variance.

its anchor bias at a median $|b|$ of 0.023 and its $\kappa > 0$ fraction at 0.84, all inside the range the other two span. SASC reaches 0.887 at k=50 against naive's 0.878 and NexCP-gated's 0.895, significant at every $k \geq 50$ ($p \leq 3 \times 10^{-5}$) and not at k=20 ($p = 0.43$), which is the same pattern the other TFMs show. Plain pooling is worse for TabDPT than for either of them, at 0.768. GAMformer is the case the criterion speaks to directly. Additivity constrains how the score distribution can move, so its median anchor bias is the smallest we measure, $|b| = 0.017$ against 0.024 for the other three TFMs. Eq. (4) then predicts a larger $k^\star$, and that is what we see: GAMformer is the only TFM for which the gate beats naive at every budget, including 0.893 against 0.865 at k=20 ($p = 2 \times 10^{-4}$), where the other three are too noisy for the gate. Its k=50 gain is also the largest, 0.898 against 0.878 ($p = 2 \times 10^{-9}$). Two consequences run the other way and we state them plainly. Its flatter source scores survive

Table 41: Source calibration size n_s (predictor held fixed). Tail coverage across the 46 concept-shift settings; nominal 0.90.

model	n_s	method	k=20	k=50	k=100	k=200
TabICLv2	250	naive	0.868	0.879	0.883	0.890
		SASC	0.868	0.890	0.894	0.896
	500	naive	0.868	0.879	0.883	0.890
		SASC	0.874	0.888	0.894	0.897
	1000	naive	0.868	0.879	0.883	0.890
		SASC	0.872	0.889	0.896	0.899
	2000	naive	0.868	0.879	0.883	0.890
		SASC	0.875	0.892	0.896	0.898
	3000	naive	0.868	0.879	0.883	0.890
		SASC	0.878	0.892	0.896	0.897
TabPFNv2	250	naive	0.870	0.878	0.882	0.888
		SASC	0.862	0.886	0.896	0.898
	500	naive	0.870	0.878	0.882	0.888
		SASC	0.867	0.886	0.895	0.897
	1000	naive	0.870	0.878	0.882	0.888
		SASC	0.866	0.888	0.897	0.898
	2000	naive	0.870	0.878	0.882	0.888
		SASC	0.866	0.889	0.896	0.899
	3000	naive	0.870	0.878	0.882	0.888
		SASC	0.866	0.889	0.897	0.899

pooling, so plain pooling reaches 0.888 and does *not* fall below naive for GAMformer, as it does for the three unconstrained TFMs. And per-bin anchoring gives it nothing: per-bin naive reaches 0.835 and per-bin anchoring 0.833, because per-bin calibration alone already repairs a conditional deficit that is the largest of any model (0.711 marginal).

Source-only calibration undercovers for all six base models. Naive few-shot repairs the mean but is unreliable in the tail at small k. SASC improves tail coverage for all six, with the ordering relative to the pooled baseline set by how flat the source scores are. The phenomenon is a property of *concept shift interacting with split conformal calibration*, so the prescription is model-agnostic: it needs no foundation model at all. Two secondary observations follow. First, the non-foundation models start from a slightly better source-only coverage on this suite, consistent with their lower source accuracy leaving less to lose. Second, the pooled baseline is less harmful for them than for the TFMs, because their source score distributions are flatter and so less damaged by pooling with a location-shifted copy. Thus the failure of pooling depends on the score geometry, whereas SASC's CV gate does not.

U Anatomy of the score-distribution shift, and a test of the theory

SASC rests on a claim about the geometry of the shift: near the $(1-\alpha)$ quantile, the target score distribution is approximately a location translate of the source's. We measure that claim, then test the two quantitative predictions of Prop. 4. Figure 26 shows the shift in the score distribution itself, and Figure 27 tests the anchor's assumption directly.

How much of the shift is location? For each setting we compute the 1-Wasserstein distance between the source and target score distributions, then recompute it after (i) the median-gap translation and (ii) also matching inter-quartile ranges. Table 47 reports the averages. A pure location correction removes 17–21% of the transport distance for four of the five TFMs and 5% for GAMformer. The added scale correction removes at most 7 points more, and for TabPFNv2, XGBoost and GAMformer it makes matters *worse*.

A 15% figure sounds like a weak endorsement of a method built on the location assumption, but it is not. The Wasserstein distance integrates the discrepancy over the *whole* distribution, including the dense central region where the two laws differ in shape and conformal calibration does not look. The anchor needs a similar *median-to-tail spread*, which is exactly the anchor bias b of Definition 1. The median of $|b|$ is ≈ 0.024 on the LAC scale, small relative to the ≈ 0.09 typical gap the anchor must correct. In other words, real tabular concept shift is *not* a pure location shift and the anchor is biased.

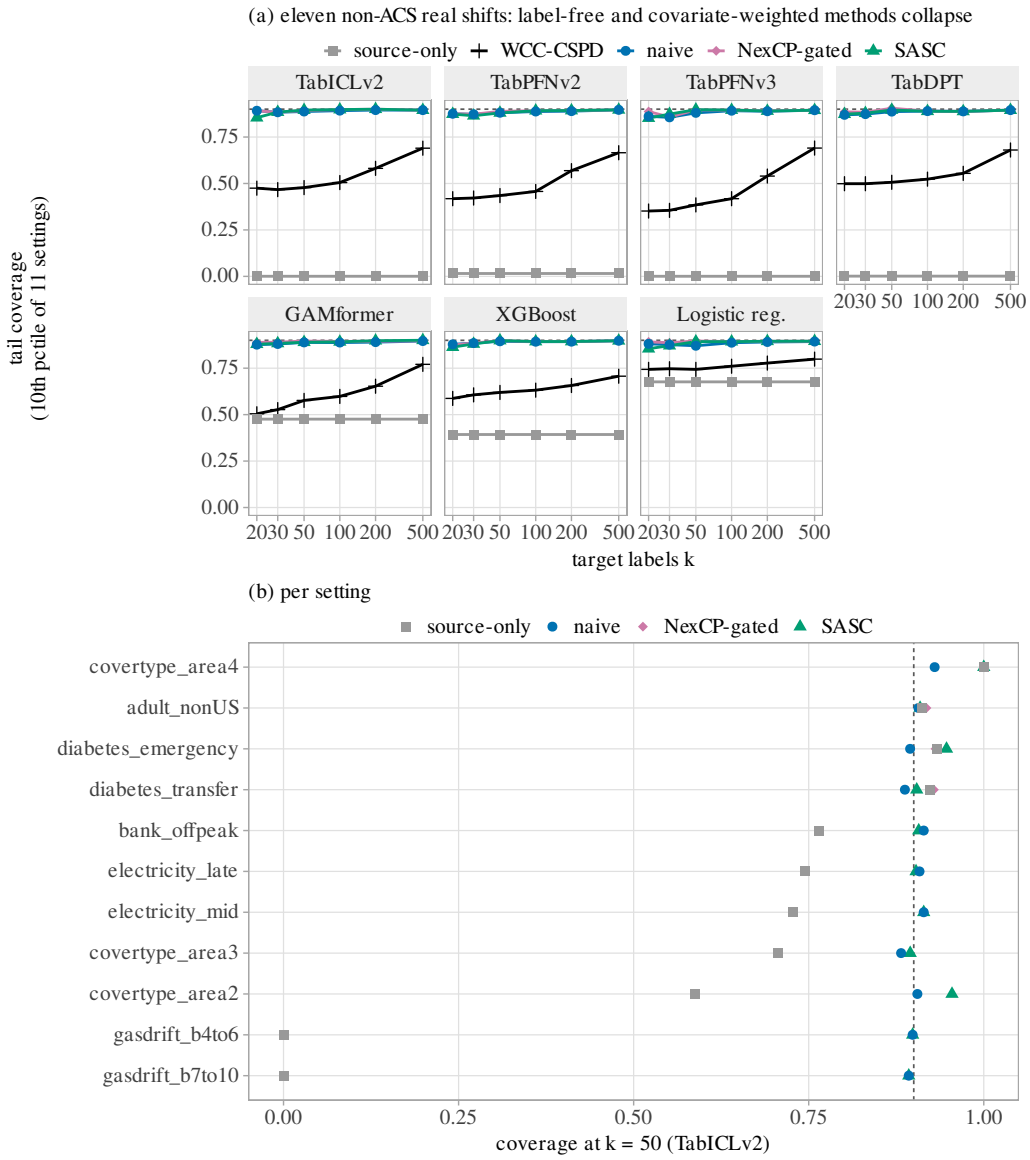


Figure 22: **The second real-shift family.** *(a)* Tail coverage across the eleven settings against budget, six base models. Source-only calibration (gray) and WCC-CSPD (black) collapse; naive few-shot (blue), NexCP-gated (pink) and SASC (green) reach nominal from k=50. *(b)* Per setting at k=50 for TabICLv2. Cover type area 4 has a single target class, so every rule covers.

SASC works anyway: the bias is small compared with the sampling noise of a k-point quantile, and the CV gate detects the settings where it is not. This also explains why the two-parameter location–scale anchor loses (§P): the scale component it spends a parameter on is not there to be corrected.

Table 47 also reports the density ratio $f_T(m_T)/f_T(q_T) \approx 2.3$–$2.6$ for LAC scores on the concept-shift suite, well above the threshold 1.67 at which κ turns positive, the precondition for the anchor to have any variance advantage (Eq. (25)).

Test 1: does the anchor's risk behave as predicted? Prop. 4 makes a falsifiable prediction about the *threshold* error, not about coverage: $\mathrm{MSE}(\hat{q}_{\mathrm{naive}})$ should fall like $1/k$, while $\mathrm{MSE}(\hat{q}_{\mathrm{anc}})$ should fall and then plateau at the bias floor b^2. Table 48 and Figure 25(a) measure both. Ground truth is the $(1-\alpha)$ quantile of the held-out target *test* scores, and we average over 20 independent subsamples of the target pool per (setting, seed, k). The prediction holds. Naive's MSE falls by a factor of 17.6

Table 42: Second real-shift family: tail coverage across eleven settings and mean set size at k=50, five base models, five seeds. Nominal 0.90.

base model	method	k=20	k=50	k=100	k=200	k=500	size$_{k=50}$
TabICLv2	source-only	0.001	0.001	0.001	0.001	0.001	0.96
	naive	0.892	0.887	0.892	0.895	0.895	1.55
	pooled	0.538	0.543	0.508	0.501	0.658	1.32
	NexCP-gated	0.892	0.893	0.898	0.900	0.895	1.58
	SASC	0.854	0.895	0.898	0.900	0.895	1.69
	SASC-cons	1.000	0.950	0.930	0.915	0.914	1.93
TabPFNv2	source-only	0.015	0.015	0.015	0.015	0.015	0.97
	naive	0.875	0.881	0.887	0.889	0.896	1.64
	pooled	0.497	0.525	0.494	0.492	0.661	1.27
	NexCP-gated	0.875	0.888	0.892	0.893	0.896	1.68
	SASC	0.873	0.880	0.892	0.893	0.896	1.73
	SASC-cons	1.000	0.952	0.931	0.918	0.913	2.04
TabPFNv3	source-only	0.000	0.000	0.000	0.000	0.000	0.97
	naive	0.861	0.880	0.892	0.889	0.894	1.56
	pooled	0.478	0.537	0.508	0.505	0.657	1.28
	NexCP-gated	0.887	0.896	0.895	0.891	0.894	1.59
	SASC	0.852	0.897	0.895	0.890	0.894	1.65
	SASC-cons	1.000	0.953	0.931	0.913	0.913	1.93
TabDPT	source-only	0.001	0.001	0.001	0.001	0.001	0.98
	naive	0.869	0.887	0.890	0.888	0.895	1.65
	pooled	0.508	0.516	0.484	0.519	0.663	1.29
	NexCP-gated	0.885	0.905	0.890	0.888	0.895	1.69
	SASC	0.873	0.896	0.889	0.888	0.895	1.73
	SASC-cons	1.000	0.953	0.932	0.917	0.914	1.99
GAMformer	source-only	0.476	0.476	0.476	0.476	0.476	1.26
	naive	0.877	0.889	0.888	0.890	0.896	2.34
	pooled	0.522	0.570	0.567	0.616	0.724	1.68
	NexCP-gated	0.889	0.896	0.894	0.895	0.897	2.36
	SASC	0.879	0.890	0.892	0.898	0.900	2.61
	SASC-cons	1.000	0.950	0.932	0.920	0.917	2.90
XGBoost	source-only	0.393	0.393	0.393	0.393	0.393	1.05
	naive	0.878	0.895	0.892	0.893	0.898	1.97
	pooled	0.481	0.508	0.507	0.507	0.669	1.49
	NexCP-gated	0.875	0.899	0.893	0.894	0.898	1.99
	SASC	0.863	0.899	0.893	0.894	0.898	2.21
	SASC-cons	1.000	0.953	0.929	0.917	0.918	2.48
Logistic reg.	source-only	0.676	0.676	0.676	0.676	0.676	1.13
	naive	0.881	0.871	0.887	0.891	0.893	1.72
	pooled	0.706	0.700	0.714	0.699	0.722	1.17
	NexCP-gated	0.892	0.897	0.893	0.895	0.895	1.77
	SASC	0.855	0.893	0.893	0.895	0.895	1.76
	SASC-cons	1.000	0.945	0.928	0.914	0.910	2.08

from k=20 to k=500 (a factor of 25 in k), while the anchor's falls by only a factor of 2 and flattens. The crossover is visible in the aggregate, where the anchor has lower risk at $k \leq 30$ (TabICL) or $k \leq 50$ (TabPFN) and higher risk after. The per-setting win rate shows it too, dropping from 0.63–0.65 at k=20 to 0.26–0.28 at k=500. SASC itself is below both estimators at $k \leq 30$, which is the shrinkage effect, but *not* beyond: at k=50 its MSE is 6.92 against naive's 5.82, and at k=500 it is 4.05 against 0.71, a factor of 5.7. The reason is visible in the selected weight. The median $w^\star$ is 0 at every budget, so the gate keeps the pure anchor, and it does so because the anchor *over*-covers in the 57% of settings with $b > 0$: an over-covering threshold passes a coverage gate by construction. The gate optimizes a one-sided coverage constraint, not threshold risk, so SASC inherits the anchor's bias floor in MSE while buying tail coverage with it. This is the same trade that §X.1 measures directly, and it is why we do not claim SASC is a minimum-risk estimator of q_T.

Table 43: Second real-shift family, per setting at k=50: coverage of source-only, naive few-shot, NexCP-gated and SASC calibration.

	TabICLv2				TabPFNv2			
setting	source	naive	NexCP	SASC	source	naive	NexCP	SASC
adult_nonUS	0.912	0.907	0.918	0.909	0.914	0.896	0.918	0.909
bank_offpeak	0.765	0.914	0.914	0.907	0.767	0.908	0.908	0.892
covertype_area2	0.588	0.905	0.905	0.955	0.575	0.905	0.905	0.953
covertype_area3	0.706	0.882	0.882	0.895	0.704	0.888	0.888	0.878
covertype_area4	1.000	0.930	1.000	1.000	1.000	0.897	1.000	1.000
diabetes_emergency	0.933	0.895	0.931	0.947	0.924	0.893	0.929	0.932
diabetes_transfer	0.923	0.887	0.929	0.904	0.927	0.881	0.933	0.886
electricity_late	0.745	0.908	0.908	0.903	0.766	0.925	0.925	0.920
electricity_mid	0.728	0.914	0.914	0.914	0.747	0.906	0.906	0.906
gasdrift_b4to6	0.001	0.898	0.898	0.898	0.015	0.919	0.919	0.919
gasdrift_b7to10	0.000	0.893	0.893	0.893	0.000	0.880	0.880	0.880

Table 44: The criterion's inputs on the second family (LAC scores): the tail density ratio, the median κ, the fraction of settings with $\kappa > 0$, and the median anchor bias $|b|$, which is ten times its ACS value.

base model	$f_T(m_T)/f_T(q_T)$	κ (median)	frac. $\kappa > 0$	median $\lvert b\rvert$
TabICLv2	2.51	+0.0367	0.75	0.2438
TabPFNv2	2.23	+0.0221	0.71	0.2897
TabPFNv3	2.44	+0.0359	0.75	0.2805
TabDPT	2.33	+0.0331	0.69	0.0917
GAMformer	2.46	+0.0405	0.69	0.2751
XGBoost	1.76	+0.0061	0.58	0.2555
Logistic reg.	2.35	+0.0391	0.82	0.3222

Test 2: does the predicted crossover $\hat{k}^\star$ match the observed one? For each setting we estimate $\hat{k}^\star = \kappa/b^2$ from the score distributions alone, and record the largest budget at which the anchor has lower threshold risk. Table 49 and Figure 25(b,c) report the Spearman correlation between the two: $\rho = +0.77$ (TabICL) and $+0.81$ (TabPFN) across the 46 settings. A plug-in asymptotic formula, using only source and target score distributions, orders the empirical crossover correctly across settings. The correlation with $1/b^2$ alone is lower ($+0.60$/$+0.66$), so the density factor κ carries real information beyond the bias.

The correlation is on the *ordering*, not the level. The median predicted $\hat{k}^\star$ is an order of magnitude larger than the median observed crossover, for two reasons. The kernel density estimate of $f_T(q_T)$ oversmooths a sparse tail and inflates κ; and b estimated from finite samples is attenuated. Both push $\hat{k}^\star$ upward. Hence $\hat{k}^\star$ is a relative diagnostic of which settings will benefit most, not an absolute rule for choosing k; the cross-validated weight serves that purpose.

What survives when only k labels are available. The validation above estimates f_T and b from the 2,500 target test scores. A practitioner has only the source scores and k labels. Table 46 re-estimates κ from the source scores (whose shape the anchor assumes is shared) and b^2 from the k labels with the bias correction of Prop. 4, and correlates the resulting $\hat{k}^\star$ with the observed crossover. The ranking survives in weakened form: $\rho = +0.45$ to $+0.69$ for TabICL and $+0.36$ to $+0.49$ for TabPFN depending on k, against $+0.77$ and $+0.81$ with test densities. The level does not survive: the bias-corrected $\hat{b}^2$ is clipped to zero for most settings at $k \leq 100$, which sends $\hat{k}^\star$ to infinity. A quantile-spacing density estimator in place of the kernel changes neither conclusion and reduces the level bias only slightly (median $\hat{k}^\star$ 891 against 999 for a median observed crossover of 100). The criterion is thus a diagnostic of *which* settings will benefit, not a budget calculator, and the main text says so.

Why the coverage gain is not the MSE gain. If we correlate $\hat{k}^\star$ with SASC's *coverage* gain over naive rather than with its threshold risk, the correlation is *negative*. There is no contradiction. Coverage gain is asymmetric, since over-covering counts as a gain, and the CV rule targets a coverage constraint rather than minimum MSE. The settings where SASC lifts coverage most are thus those where naive undercovers most, which are typically those with a large anchor bias and hence a small

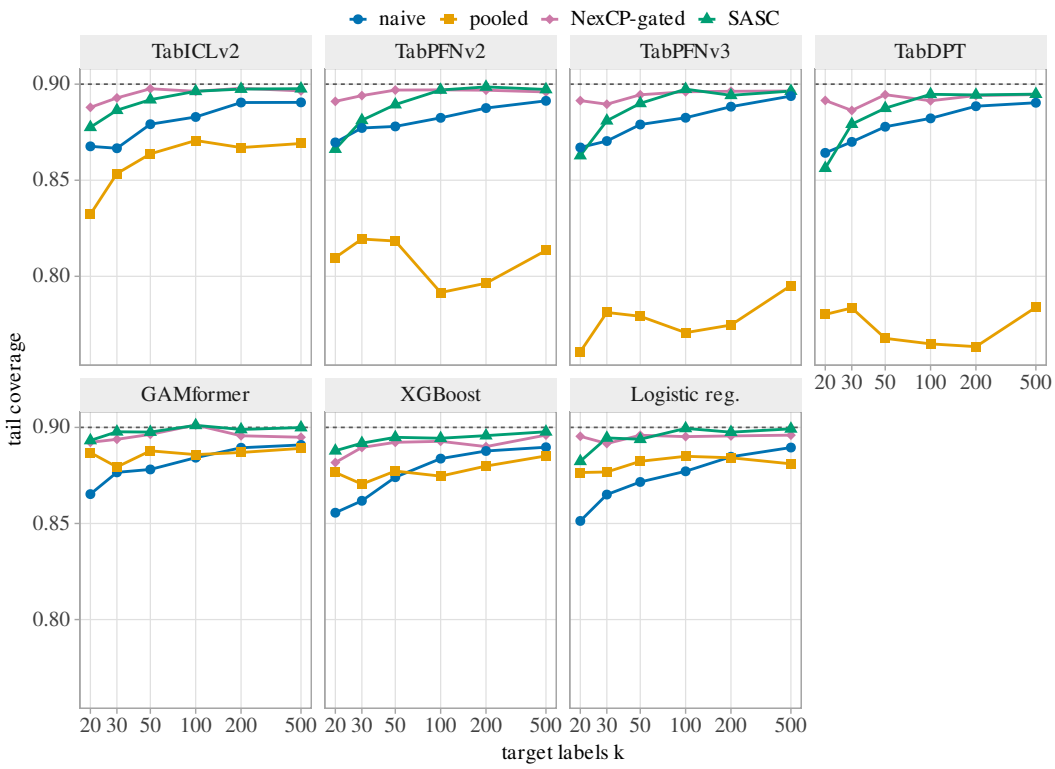


Figure 23: Tail coverage across the 46 settings for the five TFMs and two classical models under an identical protocol: SASC (green), naive few-shot (blue) and pooled (orange).

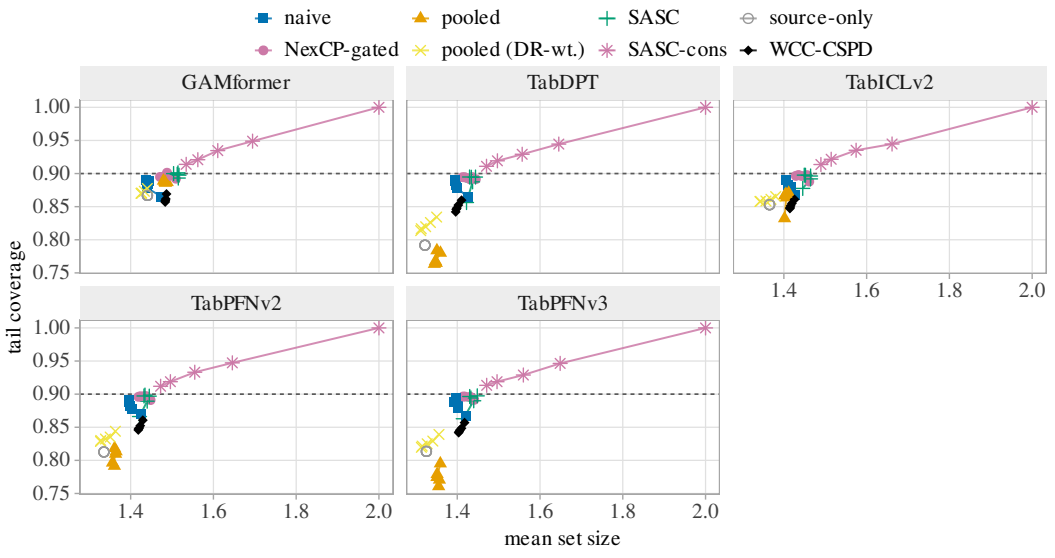


Figure 24: Tail coverage against mean set size for every calibration rule; each path runs over the budget k, and a rule is better up and to the left. SASC buys about 0.013 of tail coverage for about 3% of width; the conservative form buys a certificate for about 16%.

$\hat{k}^\star$. The MSE experiment above is the correct test of Prop. 4: it isolates estimation from the asymmetric coverage objective.

Table 45: Base-model generality: tail (10th-percentile) coverage across the 46 concept-shift settings, for two tabular foundation models and two non-foundation controls, all with identical protocol and calibration code. Nominal 0.90.

base model	method	k=20	k=50	k=100	k=200	k=500
TabICLv2	source-only	0.853	0.853	0.853	0.853	0.853
	naive	0.868	0.879	0.883	0.890	0.891
	pooled	0.832	0.864	0.871	0.867	0.869
	NexCP-gated	0.888	0.898	0.896	0.898	0.896
	SASC	0.878	0.892	0.896	0.897	0.898
TabPFNv2	source-only	0.813	0.813	0.813	0.813	0.813
	naive	0.870	0.878	0.882	0.888	0.891
	pooled	0.810	0.818	0.792	0.796	0.813
	NexCP-gated	0.891	0.897	0.897	0.897	0.896
	SASC	0.866	0.889	0.897	0.899	0.897
TabPFNv3	source-only	0.814	0.814	0.814	0.814	0.814
	naive	0.867	0.879	0.883	0.888	0.894
	pooled	0.761	0.779	0.771	0.775	0.795
	NexCP-gated	0.891	0.894	0.896	0.896	0.896
	SASC	0.863	0.890	0.897	0.894	0.896
TabDPT	source-only	0.792	0.792	0.792	0.792	0.792
	naive	0.864	0.878	0.882	0.889	0.890
	pooled	0.780	0.768	0.765	0.763	0.784
	NexCP-gated	0.892	0.895	0.891	0.894	0.895
	SASC	0.856	0.887	0.895	0.894	0.895
GAMformer	source-only	0.867	0.867	0.867	0.867	0.867
	naive	0.865	0.878	0.884	0.889	0.891
	pooled	0.887	0.888	0.886	0.887	0.889
	NexCP-gated	0.892	0.896	0.901	0.896	0.895
	SASC	0.893	0.898	0.901	0.899	0.900
XGBoost	source-only	0.858	0.858	0.858	0.858	0.858
	naive	0.856	0.874	0.884	0.888	0.890
	pooled	0.877	0.877	0.875	0.880	0.885
	NexCP-gated	0.882	0.892	0.893	0.890	0.896
	SASC	0.888	0.895	0.894	0.896	0.898
Logistic reg.	source-only	0.870	0.870	0.870	0.870	0.870
	naive	0.851	0.872	0.877	0.885	0.890
	pooled	0.877	0.882	0.885	0.884	0.881
	NexCP-gated	0.895	0.896	0.895	0.896	0.896
	SASC	0.882	0.894	0.899	0.898	0.899

U.1 Are the density-ratio weights informative?

The paper's negative result relies on an oracle gap computed with *estimated* density-ratio weights. If those weights were uninformative (the domain classifier could not tell source from target), a small oracle gap would say nothing about the shift, only about our estimator. Prop. 6 would be true but uninformative. Table 50 and Figure 28 rule this out.

For every concept-shift setting we report the domain classifier's separability (the AUC between the calibration and target weight distributions, equivalently the Mann–Whitney statistic of its scores), the normalized effective sample size, the maximum-to-mean weight ratio, and source-only coverage. The weights are informative: mean separability is 0.795 and exceeds 0.55 on 98% of settings, $n_{\text{eff}}^{\text{frac}} = 0.67$ is well below 1, and the largest weight is on average $32\times$ the mean. Real covariate movement is thus present, and the classifier detects it.

That movement does *not* explain the coverage failure. Across settings, the correlation between $n_{\text{eff}}^{\text{frac}}$ and the source-only coverage deficit is -0.29 to -0.40 depending on the base model. The sign is the expected one (more covariate shift, slightly more coverage lost) and the association is real, but it accounts for only 8–16% of the variance in the deficit. The decisive evidence remains the oracle gap of §4.2. Even the best member of the reweighting family, chosen with test labels, recovers almost

Table 46: The criterion with and without access to target test scores: Spearman correlation between $\hat{k}^\star$ and the observed crossover, and the level of each, for two density estimators.

model	densities from	b from	Spearman ρ	median $\hat{k}^\star$	median k_{obs}
TabICLv2	target test scores (kde)	test scores	+0.77	999	100
TabICLv2	source scores (kde)	k=20 labels	+0.55	∞	100
TabICLv2	source scores (kde)	k=50 labels	+0.45	∞	100
TabICLv2	source scores (kde)	k=100 labels	+0.66	∞	100
TabICLv2	source scores (kde)	k=200 labels	+0.69	∞	100
TabICLv2	source scores (kde)	k=500 labels	+0.62	∞	100
TabICLv2	target test scores (spacing)	test scores	+0.76	891	100
TabICLv2	source scores (spacing)	k=20 labels	+0.56	∞	100
TabICLv2	source scores (spacing)	k=50 labels	+0.45	∞	100
TabICLv2	source scores (spacing)	k=100 labels	+0.66	∞	100
TabICLv2	source scores (spacing)	k=200 labels	+0.63	∞	100
TabICLv2	source scores (spacing)	k=500 labels	+0.61	∞	100
TabPFNv2	target test scores (kde)	test scores	+0.81	668	100
TabPFNv2	source scores (kde)	k=20 labels	+0.36	∞	100
TabPFNv2	source scores (kde)	k=50 labels	+0.36	∞	100
TabPFNv2	source scores (kde)	k=100 labels	+0.37	∞	100
TabPFNv2	source scores (kde)	k=200 labels	+0.43	∞	100
TabPFNv2	source scores (kde)	k=500 labels	+0.42	∞	100
TabPFNv2	target test scores (spacing)	test scores	+0.82	640	100
TabPFNv2	source scores (spacing)	k=20 labels	+0.37	∞	100
TabPFNv2	source scores (spacing)	k=50 labels	+0.41	∞	100
TabPFNv2	source scores (spacing)	k=100 labels	+0.43	∞	100
TabPFNv2	source scores (spacing)	k=200 labels	+0.48	∞	100
TabPFNv2	source scores (spacing)	k=500 labels	+0.49	∞	100

Table 47: Score-shift anatomy on the 46 concept-shift settings (5 seeds). W_1 is the 1-Wasserstein distance between source and target LAC score distributions, before and after correcting location (and scale); "loc. frac." is the fraction of W_1 removed by the location correction alone. The density ratio $f_T(m_T)/f_T(q_T)$ exceeds the threshold 1.67 at which κ turns positive.

base model	W_1 raw	after loc.	after loc+scale	loc. frac.	$f_T(m_T)/f_T(q_T)$	median $\lvert b\rvert$
TabICLv2	0.0393	0.0296	0.0283	0.155	2.55	0.0245
TabPFNv2	0.0359	0.0280	0.0300	0.149	2.60	0.0237
TabPFNv3	0.0376	0.0309	0.0325	0.108	2.61	0.0286
TabDPT	0.0361	0.0281	0.0298	0.156	2.51	0.0234
GAMformer	0.0304	0.0269	0.0423	0.059	2.38	0.0175
XGBoost	0.0346	0.0289	0.0342	0.090	2.28	0.0245
Logistic reg.	0.0223	0.0168	0.0202	0.180	2.41	0.0185

none of the deficit, which is what Prop. 2 predicts if the concept term dominates and equally what a misestimated ratio would produce; §D.5 explains why we could not tell these apart. Covariate shift is thus not absent; it is present, measurable, and weakly associated with the damage. The correct statement is that correcting it does not repair the coverage, a stronger and more useful claim that five base models and 46 settings support.

V Regression: normalized residuals, CQR, and sasc

V.1 The main-text regression summary

On ACS log-income regression with absolute-residual intervals the oracle gap is 0.006 for both TFM regressors, and k=500 target labels restore coverage from 0.877–0.882 to 0.900 at a width increase from 2.1 to 2.3 target standard deviations (Table 51); AUTO-τ widens intervals without improving coverage. The problem persists under normalized-residual and CQR scores over nine target pairs,

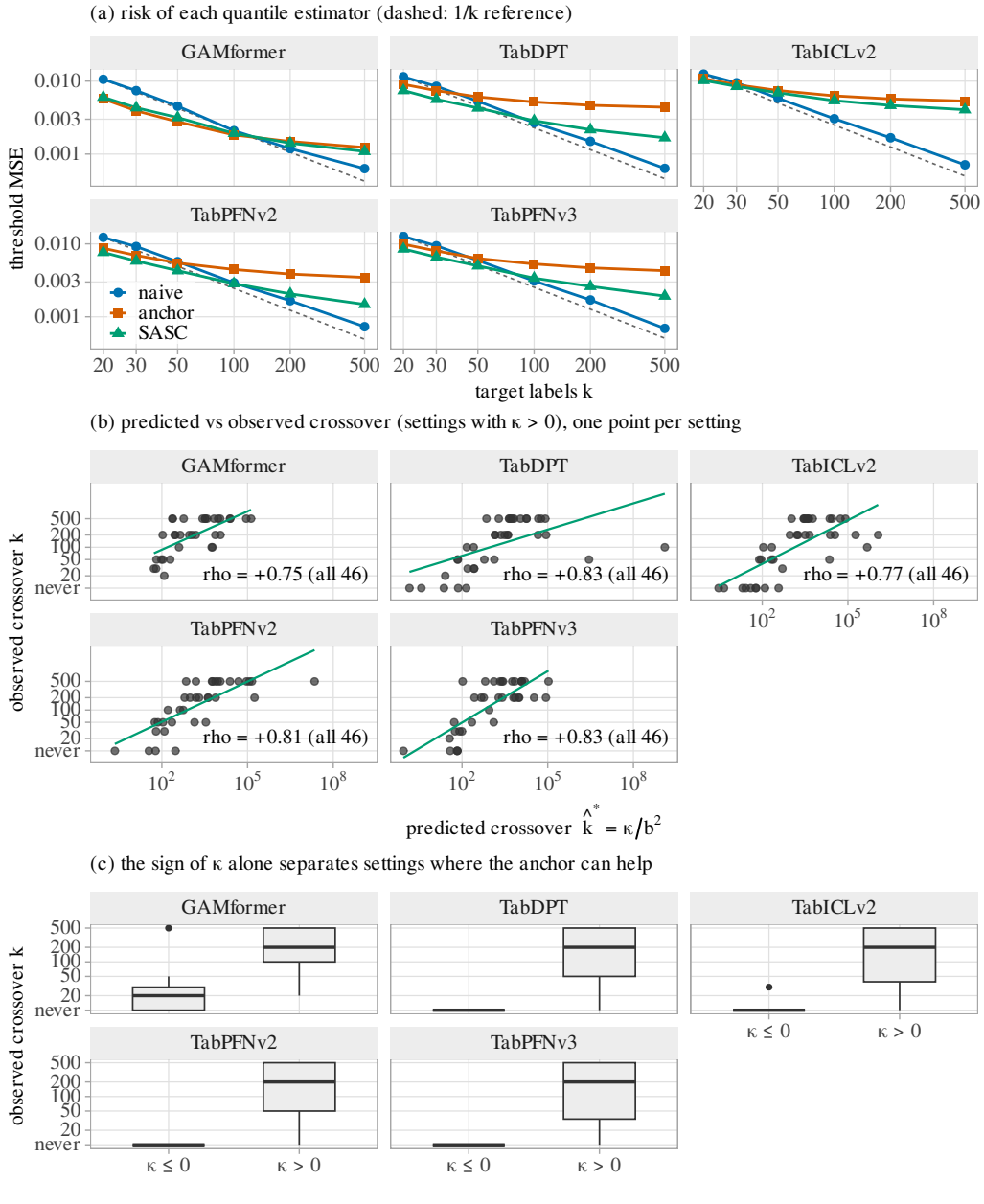


Figure 25: *(a)* Threshold mean squared error against the true target quantile by budget, averaged over 46 settings, 5 seeds and 20 resamples; dashed is the $1/k$ reference. *(b)* Predicted crossover $\hat{k}^\star = \kappa/b^2$ (Eq. (25)) against the observed crossover per setting; "never" marks settings where the anchor never has lower risk. *(c)* Crossover by the sign of κ.

where SASC improves tail coverage under all three scores with narrower intervals than naive, 2.38 against 2.44 at k=50 (Appendix V).

V.2 Main-text regression and Wine tables

Per-pair detail behind Table 51 (regression) and the Wine paragraph of §V.1, at the main text's protocol (3 seeds, absolute-residual score). §V below extends this to normalized-residual and CQR scores, nine target pairs and five seeds.

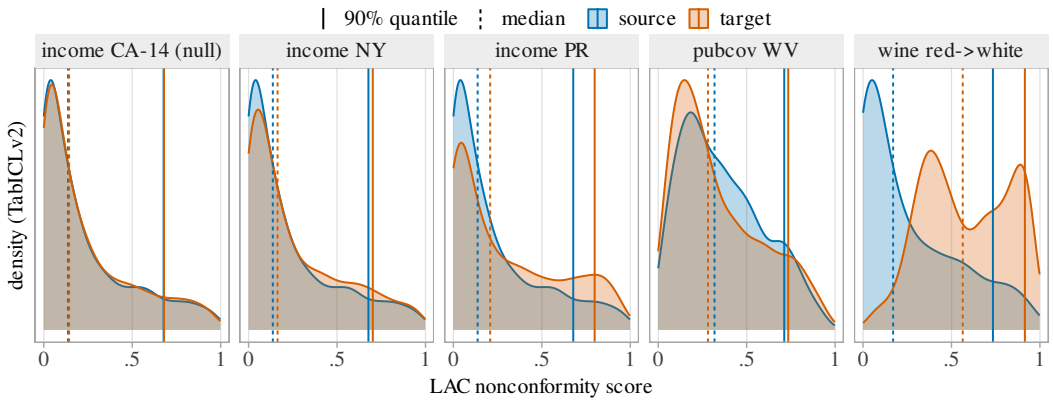


Figure 26: Kernel density estimates of the LAC score on the source calibration set (blue) and the target test set (orange) for five settings spanning the severity range, with medians (dashed) and 90% quantiles (solid).

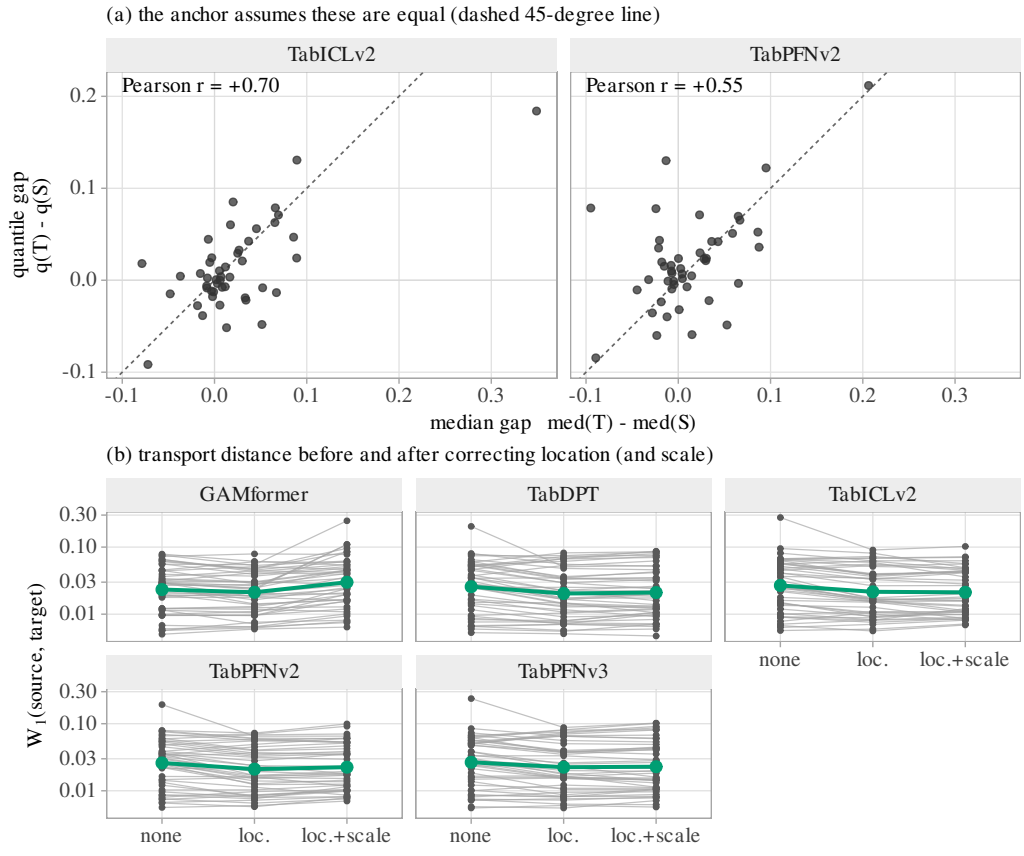


Figure 27: *(a)* Shift in the median of the score distribution against the shift in its $(1-\alpha)$ quantile, per setting; the anchor is exact on the 45° line ($r = +0.70$ TabICLv2, $+0.55$ TabPFNv2). *(b)* W_1 between source and target scores before correction, after the median translation, and after also matching inter-quartile ranges (gray: settings; green: mean).

Table 48: Test of Prop. 4: threshold mean squared error ($\times 10^3$) against the true target quantile, averaged over 46 settings, 5 seeds and 20 resamples. Naive falls as $1/k$; the anchor plateaus at its bias floor; SASC follows the anchor beyond the crossover, because its gate targets coverage rather than threshold risk.

model	estimator	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	naive $\hat{q}$	12.48	9.51	5.82	3.05	1.67	0.71
	anchor $\hat{q}$	10.80	8.92	7.43	6.28	5.69	5.33
	SASC $\hat{q}$	10.22	8.47	6.92	5.40	4.64	4.05
	frac. settings anchor wins	0.63	0.63	0.63	0.50	0.46	0.26
	mean selected w	0.18	0.21	0.30	0.34	0.39	0.42
TabPFNv2	naive $\hat{q}$	12.28	9.18	5.70	2.92	1.66	0.73
	anchor $\hat{q}$	8.70	6.95	5.46	4.45	3.85	3.47
	SASC $\hat{q}$	7.64	5.82	4.28	2.85	2.06	1.48
	frac. settings anchor wins	0.65	0.67	0.65	0.50	0.46	0.28
	mean selected w	0.20	0.26	0.35	0.39	0.44	0.48

Table 49: Predicted versus observed crossover budget. $\hat{k}^\star = \kappa/b^2$ is computed from score distributions alone (Eq. (25)); the observed crossover is the largest k at which the anchor still has lower threshold risk. ρ is Spearman's rank correlation across settings.

model	#settings	med. $\hat{k}^\star$	med. k_{obs}	$\rho(\hat{k}^\star, k_{\text{obs}})$	$\rho(1/b^2, k_{\text{obs}})$
TabICLv2	46	999	100	+0.77	+0.60
TabPFNv2	46	668	100	+0.81	+0.66
TabPFNv3	46	521	200	+0.83	+0.73
TabDPT	46	1034	100	+0.83	+0.80
GAMformer	46	260	200	+0.75	+0.45
XGBoost	46	620	100	+0.67	+0.72
Logistic reg.	46	556	100	+0.61	+0.19

Table 50: Density-ratio weight diagnostics on the 46 concept-shift settings (5 seeds). AUC is the domain classifier's separability between calibration and target (0.5 = uninformative); $w_{\max}/\bar{w}$ measures weight concentration; the last columns give source-only coverage and the correlation, across settings, between $n_{\text{eff}}^{\text{frac}}$ and the coverage deficit.

base model	domain-clf. AUC	frac. AUC>0.55	$n_{\text{eff}}^{\text{frac}}$	$w_{\max}/\bar{w}$	source-only cov.
TabICLv2	0.795	0.98	0.67	31.9	0.883
TabPFNv2	0.795	0.98	0.67	31.9	0.874
TabPFNv3	0.795	0.98	0.67	31.9	0.871
TabDPT	0.795	0.98	0.67	31.9	0.872
GAMformer	0.795	0.98	0.67	31.9	0.899
XGBoost	0.795	0.98	0.67	31.9	0.887
Logistic reg.	0.795	0.98	0.67	31.9	0.895

Correlation across settings between $n_{\text{eff}}^{\text{frac}}$ and the source-only coverage deficit: TabICLv2 -0.34, TabPFNv2 -0.39, TabPFNv3 -0.41, TabDPT -0.52, GAMformer -0.17, XGBoost -0.40, Logistic reg. -0.29. A weak association in the expected direction; it explains 8–16% of the variance.

The main text runs regression with absolute-residual intervals only. Two standard refinements exist: normalizing by a learned conditional scale, and conformalized quantile regression (Romano et al., 2019), the strongest widely used conformal regression method. Table 56 and Figure 29 test on the nine ACS log-income shift pairs whether SASC's benefit survives them, or whether a better score removes the problem SASC solves. We fit $\hat{\sigma}(x)$ by gradient boosting on an inner split of the source training data, so it never sees calibration or target labels; the conditional quantiles come from the base TFM regressor itself.

We find three things. *(i) A better score does not remove the problem.* CQR gives the tightest intervals, as designed (2.22–2.27 target standard deviations at k=50 versus 2.38–2.46 for absolute residuals). Its source-calibrated version still undercovers under shift, exactly like the absolute-residual version (0.872–0.873 tail), and its few-shot version is just as noisy at small k. The failure lies in *which*

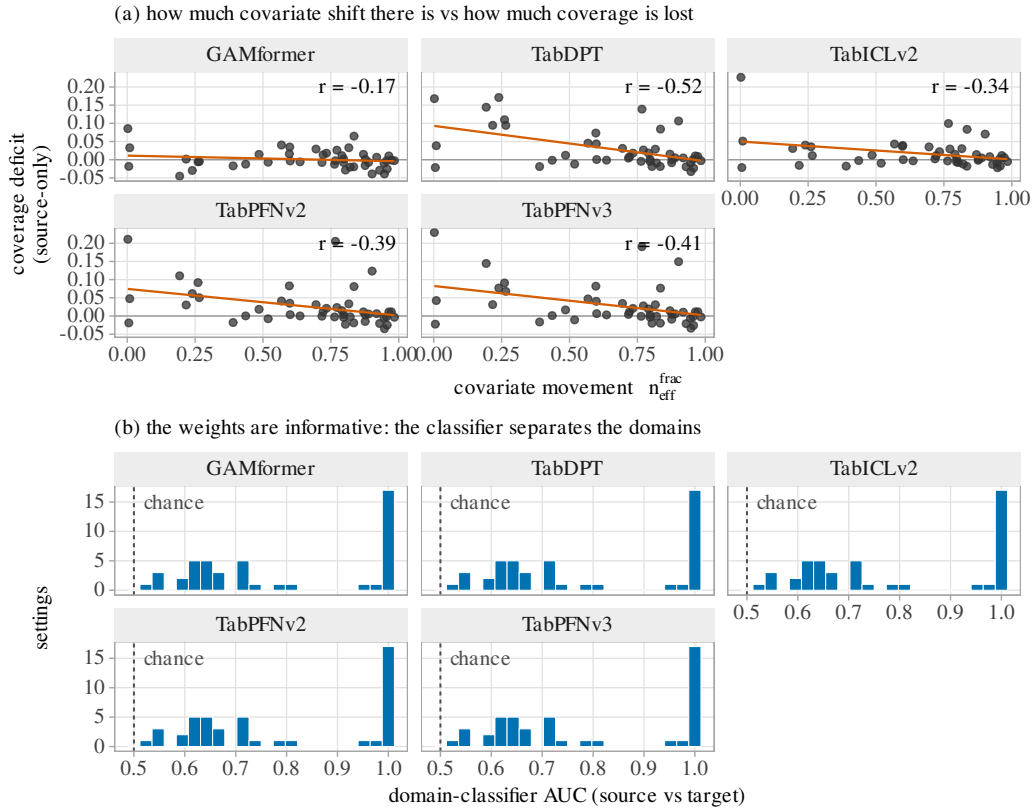


Figure 28: *(a)* Covariate movement detected by the domain classifier against the coverage lost by source-only calibration, per setting ($r = -0.34/-0.39$). *(b)* Domain-classifier AUC per setting (mean 0.795): the weights separate the domains and still cannot repair coverage.

Table 51: **ACS income regression under shift**, five pairs, 3 seeds. Coverage at nominal 0.90 and interval width in target standard deviations.

	TabICLv2		TabPFNv2	
method	cov	width	cov	width
plain	0.877	2.08	0.882	2.12
AUTO-τ	0.882	2.10	0.887	2.14
oracle	0.883	–	0.888	–
few k=500	0.900	2.25	0.901	2.28

Table 52: Regression, TabICLv2.

method	coverage@90	width (y-std)
plain	0.877	2.08
naive	0.882	2.10
auto	0.882	2.10
oracle	0.883	2.11
few k50	0.896	2.28
few k200	0.893	2.22
few k500	0.900	2.25

distribution the calibration scores come from, not in how the score is built, which is the paper's thesis, now tested against the score family designed to be adaptive. *(ii) SASC helps under all three scores.* For the absolute-residual and CQR scores it matches or beats naive's tail coverage at almost every budget for both models; the single exception is a 0.001 deficit for TabICL absolute residuals at k=100. Under the normalized score, naive is unusually conservative at k=20 and SASC sits below it there (0.882 vs 0.896, TabICL); it recovers from k=50 on. *(iii) In regression the gain is free.*

Table 53: Regression, TabPFNv2.

method	coverage@90	width (y-std)
plain	0.882	2.12
naive	0.887	2.14
auto	0.887	2.14
oracle	0.888	2.16
few k50	0.890	2.23
few k200	0.892	2.22
few k500	0.901	2.28

Table 54: Wine, TabICLv2.

method	coverage@90	size
plain	0.701	1.44
AUTO-τ	0.701	1.44
oracle	0.799	1.60
few k50	0.890	1.78
few k100	0.897	1.79
few k200	0.909	1.84
few k500	0.898	1.83

Table 55: Wine, TabPFNv2.

method	coverage@90	size
plain	0.700	1.47
AUTO-τ	0.700	1.47
oracle	0.824	1.67
few k50	0.914	1.85
few k100	0.910	1.82
few k200	0.913	1.85
few k500	0.896	1.83

Where classification costs SASC about 3–5% in set size, here it is both more reliable *and* narrower than naive (k=50, TabICL, absolute residual: 0.898 coverage at width 2.38 versus naive's 0.887 at 2.44). Residual distributions have much heavier upper tails than LAC scores, so $f_T(q_T)$ is very small, κ is large, and the anchor's variance advantage grows with it. The same criterion explains the APS/RAPS failure in §O; here it works in our favor.

One caveat: the conservative variant is expensive on the normalized score, where its width grows to 10.7–11.0 target standard deviations at k=50, because flooring at a high target order statistic multiplies through $\hat{\sigma}(x)$. The certified variant should therefore be paired with an unnormalized score, or used only at budgets where the rank inflation (Table 3) is mild.

W CLASS-CONDITIONAL CALIBRATION COMBINED WITH SASC

W.1 PER-k CLASS-CONDITIONAL TABLES FROM THE MAIN TEXT

Per-k detail behind Table 21 (right): marginal versus class-conditional (Mondrian) few-shot calibration, and the worst-class coverage each reaches. §W below adds the combination with SASC.

§4.2 shows that Mondrian (class-conditional) calibration is the one elaborate calibrator that pays for itself under concept shift, because concept shift moves the class-conditional score distributions directly. SASC and Mondrian calibration are orthogonal. One decides *which distribution* the quantile is estimated from, the other *how many* quantiles are estimated. Thus they compose: we run SASC separately within each class and anchor each class's target quantile to the same class's source scores. Table 59 and Figure 30 report the combination.

The composition helps: worst-class coverage rises from 0.884–0.891 (plain Mondrian) to 0.895–0.907 (Mondrian + SASC) across budgets and both models. The gain is stable in the budget: +0.011 at k=50, rising to +0.016 at k=500. This is *not* the pattern of the marginal case, where SASC's advantage shrinks as k grows. Mondrian calibration divides the budget, so at $k = 500$ each class still sees only a few hundred labels, and the per-class quantiles stay in the label-scarce regime where the anchor pays. The cost is a mild widening at large k (set size 1.51 vs 1.46 at k=500; at k=50 the

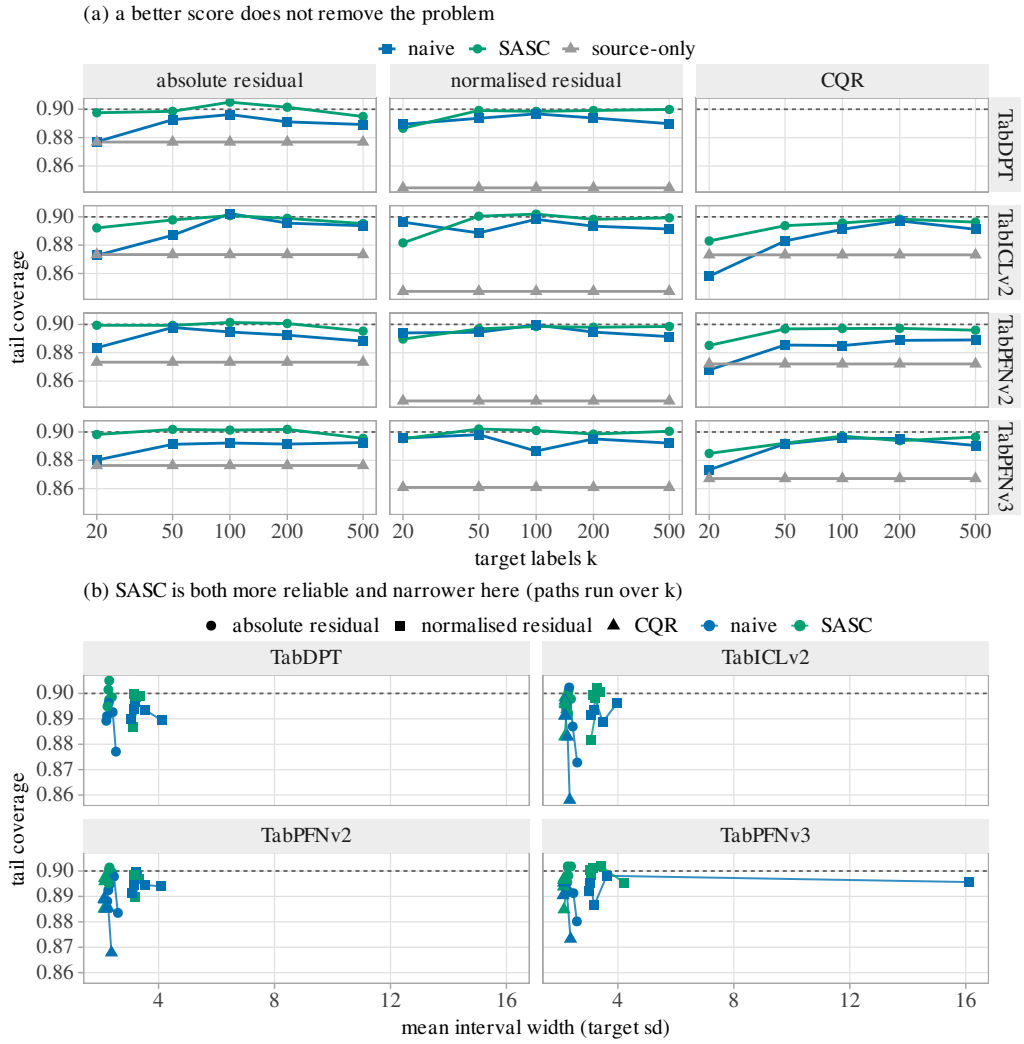


Figure 29: Regression with three nonconformity scores. *(a)* Tail coverage against budget for the absolute, normalized and CQR scores; source-calibrated intervals (gray) undercover under all three. *(b)* The same runs in the coverage/width plane.

two are within 1%). This is the natural choice for per-class reliability under shift, and it costs nothing beyond running Algorithm 1 once per class.

X Statistical significance

X.1 Matched conservatism: is the gain variance reduction or a wider set?

SASC raises tail coverage, but it also raises mean coverage and set size (Table 24). Those are not the same achievement. A rule that simply returns a higher threshold buys tail coverage with width, and any method can do that. The like-for-like control is target-only calibration at an *inflated quantile rank*: instead of the $\lceil (k+1)(1-\alpha) \rceil$ order statistic, take the $\lceil (k+1)(1-\alpha') \rceil$ one for $\alpha' < \alpha$. We sweep $\alpha' \in \{0.10, \ldots, 0.02\}$, which traces a coverage/width frontier for target-only calibration, and then read that frontier off at SASC's own mean coverage and at its own mean set size. Anything SASC contributes beyond conservatism must show up as a tail above that frontier.

The comparison also forces a second choice into the open. Our headline statistic is the 10th percentile over the 46 settings of coverage *averaged over seeds*. That measures which settings are hard. It is not

Table 56: Regression under real shift (ACS log-income, CA-2018 → nine targets, 5 seeds), $\alpha = 0.1$: tail (10th-percentile) coverage across pairs and mean interval width at k=50 in target standard deviations. Nominal coverage 0.90.

model	score	method	k=20	k=50	k=100	k=200	k=500	width$_{k=50}$
TabICLv2	absolute	source-only	0.873	0.873	0.873	0.873	0.873	2.10
		naive	0.873	0.887	0.902	0.896	0.894	2.44
		pooled	0.879	0.890	0.893	0.892	0.891	2.18
		SASC	0.892	0.898	0.901	0.899	0.895	2.38
		SASC-cons	1.000	0.968	0.943	0.919	0.911	3.94
	normalised	source-only	0.847	0.847	0.847	0.847	0.847	2.68
		naive	0.896	0.889	0.898	0.893	0.891	3.47
		pooled	0.856	0.869	0.869	0.870	0.875	2.81
		SASC	0.882	0.900	0.902	0.898	0.899	3.37
		SASC-cons	1.000	0.951	0.930	0.917	0.911	10.66
	CQR	source-only	0.873	0.873	0.873	0.873	0.873	2.07
		naive	0.858	0.883	0.891	0.897	0.891	2.27
		pooled	0.860	0.873	0.877	0.878	0.880	2.06
		SASC	0.883	0.894	0.896	0.898	0.896	2.24
		SASC-cons	1.000	0.957	0.934	0.922	0.910	3.59
TabPFNv2	absolute	source-only	0.873	0.873	0.873	0.873	0.873	2.10
		naive	0.884	0.898	0.895	0.892	0.888	2.46
		pooled	0.887	0.891	0.895	0.894	0.893	2.20
		SASC	0.899	0.899	0.901	0.901	0.895	2.40
		SASC-cons	1.000	0.962	0.940	0.918	0.908	3.88
	normalised	source-only	0.846	0.846	0.846	0.846	0.846	2.68
		naive	0.894	0.894	0.900	0.895	0.891	3.55
		pooled	0.863	0.868	0.868	0.869	0.874	2.78
		SASC	0.890	0.897	0.898	0.898	0.899	3.33
		SASC-cons	1.000	0.951	0.935	0.919	0.911	10.96
	CQR	source-only	0.872	0.872	0.872	0.872	0.872	2.02
		naive	0.868	0.885	0.885	0.889	0.889	2.27
		pooled	0.869	0.875	0.881	0.881	0.886	2.01
		SASC	0.885	0.897	0.897	0.897	0.896	2.22
		SASC-cons	1.000	0.956	0.930	0.918	0.909	3.54

Table 57: TabICLv2.

k	method	marg. cov	worst-class	size
50	marginal	0.900	0.796	1.38
50	classcond	0.917	0.892	1.55
100	marginal	0.896	0.784	1.37
100	classcond	0.905	0.881	1.47
200	marginal	0.899	0.784	1.37
200	classcond	0.902	0.882	1.45
500	marginal	0.898	0.781	1.37
500	classcond	0.900	0.885	1.43

the quantity a practitioner faces, which is the spread of realized coverage over the single calibration draw they happen to get. We therefore draw k labels 10 times independently per setting and seed and report both: the percentile over settings, and the percentile over individual draws.

The two aggregations disagree, and the disagreement is the finding. Under the percentile over settings, inflated target-only calibration matches or beats SASC everywhere, by 0.006 to 0.027 (Table 60). Averaging ten draws before taking the percentile removes exactly the draw-to-draw variance that the anchor is meant to reduce, so that aggregation cannot see the effect we claim. Under the per-draw percentile (Table 61), SASC is ahead of the size-matched frontier at k=50 by $+0.011$ to $+0.015$ for all five TFMs, and behind it by 0.002 to 0.004 at k=200. The crossover is consistent with Eq. (4), though §U explains why we do not call it a prediction: the anchor pays while labels are scarce and stops paying once they are not.

Table 58: TabPFNv2.

k	method	marg. cov	worst-class	size
50	marginal	0.901	0.789	1.39
50	classcond	0.919	0.893	1.55
100	marginal	0.898	0.778	1.38
100	classcond	0.905	0.878	1.47
200	marginal	0.901	0.783	1.38
200	classcond	0.903	0.882	1.46
500	marginal	0.899	0.778	1.37
500	classcond	0.900	0.885	1.45

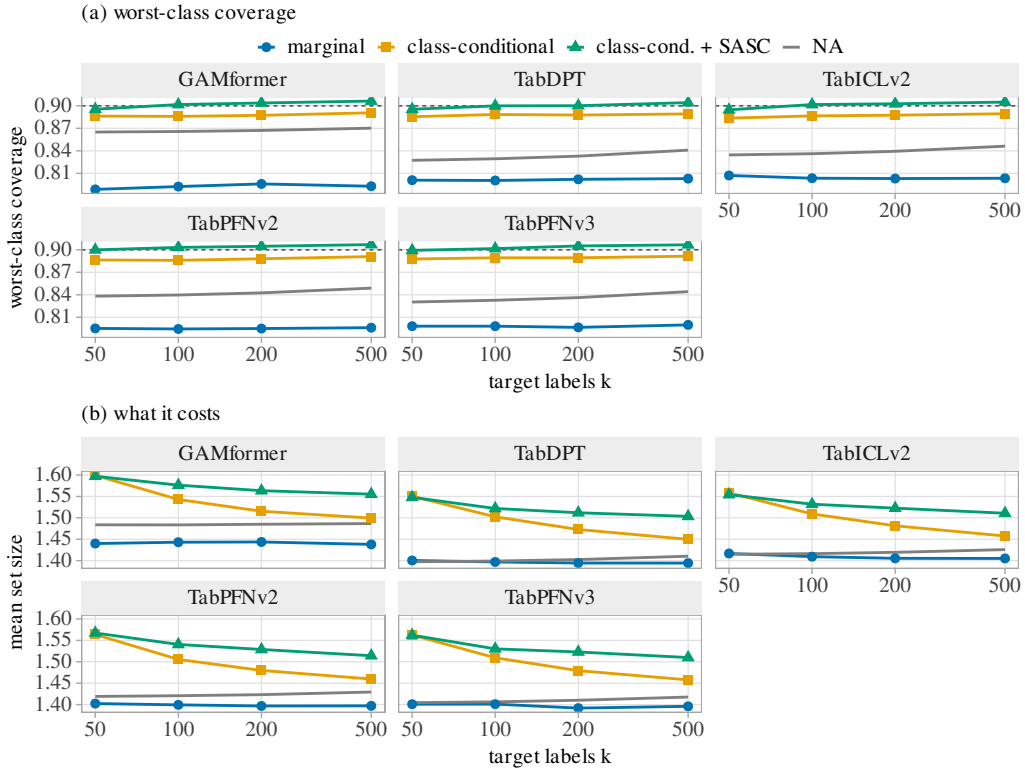


Figure 30: *(a)* Worst-class coverage for marginal, class-conditional, and class-conditional with SASC calibration. *(b)* The corresponding set sizes.

Table 59: Class-conditional calibration with and without SASC: worst-class coverage / mean set size on the 46 concept-shift settings (5 seeds). Nominal 0.90.

model	method	k=50		k=100		k=200		k=500	
		worst	size	worst	size	worst	size	worst	size
TabICLv2	marginal	0.807	1.42	0.804	1.41	0.803	1.41	0.803	1.41
	class-cond.	0.884	1.56	0.887	1.51	0.888	1.48	0.889	1.46
	class-cond. + SASC	0.895	1.55	0.902	1.53	0.903	1.52	0.905	1.51
TabPFNv2	marginal	0.795	1.40	0.794	1.40	0.795	1.40	0.796	1.40
	class-cond.	0.886	1.56	0.886	1.51	0.888	1.48	0.891	1.46
	class-cond. + SASC	0.900	1.57	0.903	1.54	0.905	1.53	0.907	1.51

Paired uncertainty, with the matching inside the resampling. The comparator is a fitted quantity: the inflated-rank frontier is estimated and then read off at SASC's mean set size. Uncertainty must therefore include the matching step, and the settings are shared across methods, seeds and calibration draws. We cluster-bootstrap over settings (2,000 replicates), recompute the frontier, the match and the difference inside every replicate, and report percentile intervals (Tables 62 and 63). Per cali-

Table 60: Matched-conservatism control, percentile over *settings*. Target-only calibration at an inflated rank, read at SASC's mean coverage and at its mean set size, is at least as good as SASC in every cell.

base model	k	SASC tail	mean cov.	size	target-only@cov	target-only@size
TabICLv2	20	0.898	0.916	1.47	**0.906**	0.900
TabICLv2	50	0.901	0.918	1.47	**0.909**	0.906
TabICLv2	100	0.899	0.917	1.46	**0.908**	0.906
TabICLv2	200	0.898	0.916	1.45	**0.909**	0.908
TabPFNv2	20	0.884	0.910	1.45	**0.897**	0.893
TabPFNv2	50	0.900	0.915	1.45	**0.906**	0.903
TabPFNv2	100	0.897	0.913	1.44	**0.903**	0.903
TabPFNv2	200	0.897	0.913	1.44	**0.903**	0.903
TabPFNv3	20	0.880	0.909	1.44	**0.894**	0.891
TabPFNv3	50	0.896	0.915	1.45	**0.906**	0.903
TabPFNv3	100	0.895	0.913	1.44	**0.906**	0.905
TabPFNv3	200	0.895	0.913	1.43	**0.905**	0.904
TabDPT	20	0.868	0.908	1.44	**0.896**	0.893
TabDPT	50	0.895	0.914	1.45	**0.905**	0.903
TabDPT	100	0.895	0.913	1.44	**0.905**	0.904
TabDPT	200	0.895	0.913	1.43	**0.903**	0.903
GAMformer	20	0.906	0.923	1.53	**0.915**	0.908
GAMformer	50	0.907	0.927	1.52	**0.917**	0.912
GAMformer	100	0.900	0.924	1.51	**0.916**	0.912
GAMformer	200	0.901	0.924	1.51	**0.916**	0.913

Table 61: The same control, percentile over individual *calibration draws*, which is the deployment distribution. Here SASC is ahead at k=50 for all five TFMs and behind by k=200.

base model	k	SASC tail	mean cov.	size	target-only@cov	target-only@size
TabICLv2	20	**0.857**	0.916	1.47	0.837	0.829
TabICLv2	50	**0.874**	0.918	1.47	0.866	0.863
TabICLv2	100	0.878	0.917	1.46	**0.879**	0.877
TabICLv2	200	0.885	0.916	1.45	**0.890**	0.888
TabPFNv2	20	**0.851**	0.910	1.45	0.827	0.822
TabPFNv2	50	**0.872**	0.915	1.45	0.860	0.857
TabPFNv2	100	**0.877**	0.913	1.44	0.875	0.874
TabPFNv2	200	0.884	0.913	1.44	**0.886**	0.886
TabPFNv3	20	**0.848**	0.909	1.44	0.824	0.819
TabPFNv3	50	**0.869**	0.915	1.45	0.861	0.858
TabPFNv3	100	**0.875**	0.913	1.44	0.874	0.874
TabPFNv3	200	0.883	0.913	1.43	**0.886**	0.886
TabDPT	20	**0.844**	0.908	1.44	0.824	0.820
TabDPT	50	**0.871**	0.914	1.45	0.860	0.857
TabDPT	100	**0.876**	0.913	1.44	0.874	0.873
TabDPT	200	0.882	0.913	1.43	**0.885**	0.885
GAMformer	20	**0.864**	0.923	1.53	0.848	0.838
GAMformer	50	**0.883**	0.927	1.52	0.876	0.870
GAMformer	100	0.886	0.924	1.51	**0.888**	0.883
GAMformer	200	0.893	0.924	1.51	**0.899**	0.895

bration draw the gain holds up where it exists: the interval excludes zero for every one of the five TFMs at k=20, 30 and 50 (TabICL $+0.028$ $[+0.018, +0.037]$, $+0.021$ $[+0.013, +0.028]$, $+0.011$ $[+0.004, +0.017]$; TabPFNv3 $+0.029$, $+0.018$, $+0.011$), includes zero at k=100, and lies below zero at k=500 for all five. Per setting the difference is negative at every budget and significantly so from k=100.

Table 62: Marginal coverage: SASC minus size-matched inflated-rank target-only, 10th-percentile coverage, with 95% paired cluster-bootstrap intervals over settings. Bold: interval excludes zero.

base model	tail	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	per draw	**+0.028 [+0.018,+0.037]**	**+0.021 [+0.013,+0.028]**	**+0.011 [+0.004,+0.017]**	+0.001 [-0.005,+0.007]	-0.004 [-0.009,+0.001]	-0.008 [-0.014,-0.003]
TabICLv2	per setting	-0.002 [-0.011,+0.005]	-0.003 [-0.010,+0.006]	-0.005 [-0.011,+0.001]	-0.007 [-0.014,-0.002]	-0.009 [-0.014,-0.004]	-0.008 [-0.015,-0.003]
TabPFNv2	per draw	**+0.029 [+0.014,+0.038]**	**+0.021 [+0.013,+0.029]**	**+0.015 [+0.008,+0.021]**	+0.003 [-0.002,+0.008]	-0.002 [-0.006,+0.003]	-0.005 [-0.010,-0.001]
TabPFNv2	per setting	-0.009 [-0.022,+0.003]	-0.005 [-0.012,+0.004]	-0.004 [-0.011,+0.001]	-0.006 [-0.012,-0.001]	-0.005 [-0.010,-0.001]	-0.005 [-0.012,-0.001]
TabPFNv3	per draw	**+0.029 [+0.016,+0.039]**	**+0.018 [+0.009,+0.027]**	**+0.011 [+0.005,+0.017]**	+0.002 [-0.003,+0.007]	-0.003 [-0.008,+0.002]	-0.006 [-0.012,-0.002]
TabPFNv3	per setting	-0.010 [-0.022,-0.001]	-0.008 [-0.017,+0.002]	-0.008 [-0.013,-0.001]	-0.010 [-0.014,-0.004]	-0.010 [-0.014,-0.004]	-0.010 [-0.014,-0.004]
TabDPT	per draw	**+0.024 [+0.002,+0.035]**	**+0.019 [+0.010,+0.027]**	**+0.013 [+0.007,+0.019]**	+0.003 [-0.002,+0.008]	-0.003 [-0.008,+0.002]	-0.010 [-0.014,-0.004]
TabDPT	per setting	-0.024 [-0.036,-0.003]	-0.007 [-0.020,+0.003]	-0.008 [-0.013,-0.001]	-0.009 [-0.013,-0.004]	-0.008 [-0.012,-0.002]	-0.007 [-0.012,-0.002]
GAMformer	per draw	**+0.026 [+0.015,+0.037]**	**+0.022 [+0.015,+0.029]**	**+0.013 [+0.008,+0.020]**	+0.003 [-0.003,+0.007]	-0.002 [-0.008,+0.004]	-0.008 [-0.014,-0.002]
GAMformer	per setting	-0.002 [-0.012,+0.005]	-0.004 [-0.013,+0.002]	-0.005 [-0.012,+0.001]	-0.011 [-0.017,-0.006]	-0.011 [-0.017,-0.004]	-0.013 [-0.018,-0.006]

Table 63: Conditional coverage: per-bin SASC minus size-matched per-bin inflated-rank target-only, worst-bin coverage, same intervals.

base model	tail	k=50	k=100	k=200	k=500
TabICLv2	per draw	+0.022 [-0.006,+0.059]	**+0.013 [+0.002,+0.026]**	**+0.013 [+0.002,+0.030]**	-0.001 [-0.006,+0.005]
TabICLv2	per setting	+0.029 [-0.005,+0.063]	+0.005 [-0.010,+0.023]	+0.005 [-0.002,+0.021]	-0.003 [-0.009,+0.007]
TabPFNv2	per draw	**+0.029 [+0.011,+0.060]**	**+0.026 [+0.009,+0.044]**	+0.008 [-0.005,+0.024]	-0.005 [-0.014,+0.005]
TabPFNv2	per setting	+0.026 [-0.002,+0.034]	+0.013 [-0.000,+0.026]	+0.001 [-0.009,+0.016]	-0.003 [-0.010,+0.006]
TabPFNv3	per draw	**+0.043 [+0.002,+0.095]**	**+0.024 [+0.004,+0.050]**	+0.008 [-0.003,+0.021]	-0.001 [-0.008,+0.007]
TabPFNv3	per setting	**+0.026 [+0.002,+0.058]**	+0.004 [-0.015,+0.022]	+0.000 [-0.021,+0.016]	-0.004 [-0.010,+0.004]
TabDPT	per draw	**+0.030 [+0.008,+0.077]**	**+0.028 [+0.007,+0.051]**	+0.004 [-0.004,+0.015]	-0.005 [-0.013,+0.004]
TabDPT	per setting	**+0.024 [+0.002,+0.042]**	+0.013 [-0.002,+0.026]	+0.006 [-0.006,+0.014]	-0.003 [-0.008,+0.021]
GAMformer	per draw	+0.037 [-0.009,+0.067]	+0.013 [-0.007,+0.030]	+0.004 [-0.001,+0.015]	-0.000 [-0.007,+0.010]
GAMformer	per setting	-0.002 [-0.042,+0.037]	+0.004 [-0.005,+0.019]	-0.004 [-0.011,+0.010]	-0.000 [-0.007,+0.007]

Sensitivity to the resampling unit. Settings are not the only dependence. The 45 ACS settings share five tasks and one source population, and the paper's significance tests elsewhere use task-level blocks for that reason. We therefore repeat the bootstrap with the six tasks (five ACS tasks and Wine) as the resampling unit, again recomputing the frontier and the match inside every replicate (Tables 64 and 65). With six blocks the intervals are necessarily wider, and the picture coarsens rather than reverses. Per draw at k=50 the interval still excludes zero for TabPFNv2 ($+0.015$ $[+0.002, +0.023]$), TabPFNv3 ($+0.011$ $[+0.000, +0.019]$), TabDPT ($+0.013$ $[+0.005, +0.021]$) and GAMformer ($+0.013$ $[+0.002, +0.021]$) but not TabICL ($+0.011$ $[-0.002, +0.020]$); at k=20 only TabICL's excludes it ($+0.028$ $[+0.002, +0.040]$), and at k=30 TabICL, TabPFNv3 and GAMformer. No task-block interval at $k \leq 50$ lies below zero, and at k=500 two of four lie entirely below it. Per setting every point estimate stays negative and eight of the 24 intervals exclude zero from above. For the conditional comparison the k=50 intervals exclude zero for TabICL, TabPFNv3 and TabDPT under task blocks, against TabPFNv3 and TabDPT under setting blocks, and include it for TabPFNv2 and GAMformer under both. We read this as follows: the per-draw gain at $k \leq 50$ is a property of this benchmark that survives the coarser resampling for most base models, and its generalization beyond these five census tasks is exactly as uncertain as six blocks make it.

Table 64: Marginal comparison of Table 62 with *tasks* as the bootstrap blocks (6 blocks, 2,000 replicates).

base model	tail	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	per draw	**+0.028 [+0.002,+0.040]**	**+0.021 [+0.001,+0.033]**	+0.011 [-0.002,+0.020]	+0.001 [-0.011,+0.009]	-0.004 [-0.012,+0.004]	-0.008 [-0.017,-0.001]
TabICLv2	per setting	-0.002 [-0.006,+0.002]	-0.003 [-0.010,+0.005]	-0.005 [-0.009,+0.002]	-0.007 [-0.013,+0.000]	-0.009 [-0.014,-0.002]	-0.008 [-0.014,-0.004]
TabPFNv2	per draw	+0.029 [-0.012,+0.042]	+0.021 [-0.002,+0.033]	**+0.015 [+0.002,+0.023]**	+0.003 [-0.004,+0.009]	-0.002 [-0.006,+0.003]	-0.005 [-0.010,+0.000]
TabPFNv2	per setting	-0.009 [-0.024,+0.005]	-0.005 [-0.013,+0.006]	-0.004 [-0.008,+0.003]	-0.006 [-0.012,+0.002]	-0.005 [-0.010,+0.001]	-0.005 [-0.011,+0.001]
TabPFNv3	per draw	+0.029 [-0.020,+0.045]	**+0.018 [+0.001,+0.030]**	**+0.011 [+0.000,+0.019]**	+0.002 [-0.005,+0.008]	-0.003 [-0.008,+0.002]	-0.006 [-0.012,-0.002]
TabPFNv3	per setting	-0.010 [-0.030,+0.005]	-0.008 [-0.019,+0.005]	-0.008 [-0.014,-0.000]	-0.010 [-0.015,-0.001]	-0.010 [-0.015,-0.002]	-0.010 [-0.015,-0.002]
TabDPT	per draw	+0.024 [-0.041,+0.042]	+0.019 [-0.006,+0.032]	**+0.013 [+0.005,+0.021]**	+0.003 [-0.002,+0.008]	-0.003 [-0.008,+0.004]	-0.010 [-0.014,-0.003]
TabDPT	per setting	-0.024 [-0.039,+0.006]	-0.007 [-0.021,+0.006]	-0.008 [-0.012,-0.001]	-0.009 [-0.013,-0.002]	-0.008 [-0.013,-0.001]	-0.007 [-0.013,+0.000]
GAMformer	per draw	+0.026 [-0.001,+0.041]	**+0.022 [+0.006,+0.032]**	**+0.013 [+0.002,+0.021]**	+0.003 [-0.008,+0.009]	-0.002 [-0.015,+0.006]	-0.008 [-0.020,+0.001]
GAMformer	per setting	-0.002 [-0.009,+0.003]	-0.004 [-0.015,+0.007]	-0.005 [-0.014,+0.000]	-0.011 [-0.025,-0.003]	-0.011 [-0.020,-0.002]	-0.013 [-0.022,-0.004]

We state the consequence plainly. *The marginal-coverage gain reported in §4.1 is mostly conservatism.* Against a target-only rule tuned to the same width it is worth $+0.02$ to $+0.03$ per draw at

Table 65: Conditional comparison of Table 63 with tasks as the bootstrap blocks.

base model	tail	k=50	k=100	k=200	k=500
TabICLv2	per draw	**+0.022 [+0.011,+0.065]**	+0.013 [-0.006,+0.033]	+0.013 [-0.002,+0.037]	-0.001 [-0.008,+0.007]
TabICLv2	per setting	**+0.029 [+0.003,+0.102]**	+0.005 [-0.008,+0.059]	+0.005 [-0.004,+0.054]	-0.003 [-0.014,+0.023]
TabPFNv2	per draw	**+0.029 [+0.007,+0.068]**	**+0.026 [+0.004,+0.050]**	+0.008 [-0.024,+0.026]	-0.005 [-0.022,+0.007]
TabPFNv2	per setting	+0.026 [-0.004,+0.032]	+0.013 [-0.006,+0.036]	+0.001 [-0.017,+0.026]	-0.003 [-0.013,+0.007]
TabPFNv3	per draw	**+0.043 [+0.019,+0.126]**	+0.024 [-0.006,+0.068]	+0.008 [-0.008,+0.028]	-0.001 [-0.010,+0.009]
TabPFNv3	per setting	**+0.026 [+0.007,+0.096]**	+0.004 [-0.015,+0.045]	+0.000 [-0.024,+0.030]	-0.004 [-0.012,+0.009]
TabDPT	per draw	**+0.030 [+0.015,+0.107]**	**+0.028 [+0.003,+0.071]**	+0.004 [-0.008,+0.017]	-0.005 [-0.025,+0.006]
TabDPT	per setting	**+0.024 [+0.002,+0.060]**	**+0.013 [+0.001,+0.056]**	+0.006 [-0.006,+0.019]	-0.003 [-0.016,+0.045]
GAMformer	per draw	+0.037 [-0.001,+0.067]	+0.013 [-0.012,+0.038]	+0.004 [-0.005,+0.017]	-0.000 [-0.016,+0.015]
GAMformer	per setting	-0.002 [-0.027,+0.010]	+0.004 [-0.005,+0.020]	-0.004 [-0.012,+0.012]	-0.000 [-0.009,+0.007]

$k \leq 30$ and about $+0.01$ at k=50, with intervals that exclude zero, and nothing beyond. Readers who care about the tail of setting-averaged coverage should use rank inflation, which is simpler and has an exact finite-sample interpretation.

Table 66: Matched conservatism for *conditional* coverage: per-bin anchoring against per-bin target-only calibration at an inflated rank, worst confidence-bin coverage.

base model	k	SASC/bin	mean cov.	target-only/bin@cov	@size
TabICLv2	50	**0.835**	0.923	0.806	0.806
TabICLv2	100	**0.854**	0.922	0.853	0.848
TabICLv2	200	**0.869**	0.919	0.868	0.864
TabPFNv2	50	**0.846**	0.925	0.821	0.820
TabPFNv2	100	**0.854**	0.922	0.848	0.841
TabPFNv2	200	0.865	0.920	**0.870**	0.865
TabPFNv3	50	**0.836**	0.920	0.810	0.810
TabPFNv3	100	0.845	0.920	**0.848**	0.841
TabPFNv3	200	0.863	0.918	**0.867**	0.863
TabDPT	50	**0.843**	0.925	0.820	0.819
TabDPT	100	**0.856**	0.924	0.850	0.843
TabDPT	200	**0.875**	0.920	0.872	0.869
GAMformer	50	0.833	0.925	**0.837**	0.835
GAMformer	100	0.859	0.929	**0.865**	0.855
GAMformer	200	0.870	0.927	**0.883**	0.874

The conditional result is less settled than we first wrote. With three confidence bins, k=50 leaves about 17 labels per bin, and there per-bin anchoring beats per-bin inflation by $+0.024$ to $+0.029$ in the point estimate for TabICL, TabPFNv2, TabPFNv3 and TabDPT (Table 66); GAMformer is at -0.002. But the worst-bin statistic is a 10th percentile over only 46 settings, and its paired intervals include zero for TabICL ($[-0.005, +0.063]$) and TabPFNv2 ($[-0.002, +0.034]$); only TabPFNv3's ($[+0.002, +0.058]$) and TabDPT's ($[+0.002, +0.042]$) exclude it. By k=200 every interval straddles zero. We therefore describe the conditional gain as suggestive and consistent in sign, not as established, and it is the marginal per-draw result above that carries the paired evidence. We note the coincidence with the crossover budget of Eq. (4) but do not claim it as a prediction. Eq. (4) compares the *pure anchor* with the *target-only quantile* in squared error, marginally; what is measured here is a coverage-gated blend against a rank-inflated comparator, within confidence bins. We have not derived a crossover for that quantity, and §U reports that $\hat{k}^\star$ correlates negatively with SASC's coverage gain, so the two should not be conflated. The defensible statement is empirical: the per-bin advantage is present while the bin is label-scarce and absent once it is not.

Table 69 and Figure 31 report, for every base model and budget, bootstrap 95% confidence intervals for the tail (10th-percentile) coverage of naive and SASC (2000 resamples over settings), the fraction of settings in which SASC reaches higher coverage, and the p-value of a paired Wilcoxon signed-rank test across the 46 settings.

The improvement is statistically significant ($p < 10^{-4}$) for every base model at every budget $k \geq 50$, and SASC wins in 72–89% of individual settings. The two exceptions are at $k = 20$: the TabPFN comparison is not significant ($p = 0.43$, and the point estimate slightly favors naive), and the TabICL comparison is only marginal ($p = 0.02$). This matches the rest of the paper: at $k = 20$ the five-fold gate has too few points to reject a biased anchor (§D.4 quantifies the acceptance probability), and the 50k-scale experiment (§G) shows a residual gap in the same regime. We therefore claim that SASC improves tail reliability significantly from $k \approx 30$ upward and is statistically indistinguishable from naive below that.

Dependence among settings. The 46 settings share one source and five tasks, so the paired test above treats dependent settings as independent. Table 68 therefore resamples whole ACS tasks (Wine as its own block) and recomputes the difference in tail coverage 4,000 times. The gated rules' gains over naive survive: at k=50 the block intervals exclude zero for NexCP-gated on all five TFMs and for SASC on four, TabDPT's lower bound sitting at zero. Plain pooling's deficit does not, because a few settings of one task drive it. Table 67 reports the 25th percentile and the mean alongside the 10th percentile, which is the fifth-smallest of 46 values.

Table 67: Tail, 25th-percentile and mean coverage at k=50 with the block-bootstrap interval for the difference in tail coverage against naive.

model	rule	worst (p10)	p25	mean	size	Δp10 vs naive [95% block CI]	p_{block}
TabICLv2	naive	0.879	0.887	0.900	1.42	–	–
	pooled	0.864	0.884	0.897	1.41	-0.015 [-0.093, +0.013]	0.653
	WCC-CSPD	0.844	0.864	0.873	1.39	-0.031 [-0.047, -0.001]	0.025
	NexCP-gated	0.898	0.906	0.914	1.46	+0.018 [+0.007, +0.029]	0.000
	SASC	0.892	0.904	0.917	1.46	+0.013 [+0.003, +0.028]	0.005
TabPFNv2	naive	0.878	0.888	0.898	1.40	–	–
	pooled	0.818	0.879	0.882	1.36	-0.060 [-0.152, +0.006]	0.506
	WCC-CSPD	0.841	0.865	0.872	1.39	-0.030 [-0.065, +0.004]	0.098
	NexCP-gated	0.897	0.902	0.911	1.44	+0.019 [+0.005, +0.024]	0.009
	SASC	0.889	0.899	0.912	1.44	+0.011 [+0.003, +0.017]	0.000

The 10th percentile of 46 settings is the 5th-smallest value; block bootstrap resamples the five ACS tasks (Wine as its own block), 4000 draws.

Table 68: Block-bootstrap 95% intervals over tasks for the gain in tail coverage over naive, all budgets and base models.

model	rule	k=20	k=30	k=50	k=100	k=200	k=500
TabICLv2	SASC	+0.010 [-0.013,+0.030]	+0.020 [+0.007,+0.029]	+0.013 [+0.003,+0.028]	+0.013 [+0.011,+0.021]	+0.007 [+0.006,+0.009]	+0.007 [+0.003,+0.015]
TabICLv2	NexCP-gated	+0.020 [+0.012,+0.034]	+0.026 [+0.012,+0.037]	+0.018 [+0.007,+0.029]	+0.013 [+0.004,+0.025]	+0.007 [+0.001,+0.013]	+0.006 [+0.002,+0.011]
TabPFNv2	SASC	-0.003 [-0.039,+0.027]	+0.004 [-0.005,+0.024]	+0.011 [+0.003,+0.017]	+0.014 [+0.009,+0.021]	+0.011 [+0.007,+0.021]	+0.006 [+0.001,+0.008]
TabPFNv2	NexCP-gated	+0.021 [-0.001,+0.042]	+0.017 [+0.008,+0.024]	+0.019 [+0.005,+0.024]	+0.015 [+0.011,+0.024]	+0.009 [+0.003,+0.023]	+0.005 [+0.000,+0.008]
TabPFNv3	SASC	-0.004 [-0.037,+0.024]	+0.011 [-0.014,+0.026]	+0.011 [+0.004,+0.016]	+0.015 [+0.009,+0.022]	+0.006 [+0.002,+0.017]	+0.003 [+0.002,+0.004]
TabPFNv3	NexCP-gated	+0.024 [+0.019,+0.033]	+0.019 [+0.007,+0.029]	+0.015 [+0.004,+0.022]	+0.014 [+0.004,+0.021]	+0.008 [+0.002,+0.021]	+0.003 [+0.001,+0.005]
TabDPT	SASC	-0.008 [-0.038,+0.031]	+0.009 [-0.023,+0.027]	+0.010 [+0.000,+0.017]	+0.013 [+0.005,+0.019]	+0.006 [+0.004,+0.009]	+0.004 [+0.000,+0.007]
TabDPT	NexCP-gated	+0.027 [+0.005,+0.045]	+0.016 [-0.002,+0.037]	+0.017 [+0.007,+0.027]	+0.009 [-0.001,+0.022]	+0.006 [+0.002,+0.011]	+0.004 [-0.001,+0.006]
GAMformer	SASC	+0.028 [+0.013,+0.044]	+0.021 [+0.011,+0.028]	+0.019 [+0.015,+0.029]	+0.017 [+0.005,+0.026]	+0.010 [+0.005,+0.017]	+0.009 [+0.006,+0.014]
GAMformer	NexCP-gated	+0.027 [+0.016,+0.043]	+0.017 [+0.001,+0.025]	+0.018 [+0.011,+0.028]	+0.017 [+0.004,+0.024]	+0.006 [-0.000,+0.022]	+0.004 [+0.000,+0.014]
XGBoost	SASC	+0.032 [+0.003,+0.044]	+0.030 [+0.009,+0.041]	+0.021 [+0.014,+0.027]	+0.011 [+0.002,+0.017]	+0.008 [+0.001,+0.017]	+0.008 [+0.004,+0.013]
XGBoost	NexCP-gated	+0.026 [+0.004,+0.063]	+0.028 [+0.006,+0.051]	+0.018 [+0.011,+0.025]	+0.009 [+0.005,+0.017]	+0.002 [-0.001,+0.015]	+0.006 [-0.001,+0.011]
Logistic reg.	SASC	+0.031 [+0.012,+0.043]	+0.029 [+0.014,+0.046]	+0.022 [+0.014,+0.034]	+0.022 [+0.015,+0.027]	+0.013 [+0.006,+0.020]	+0.010 [+0.004,+0.013]
Logistic reg.	NexCP-gated	+0.044 [+0.020,+0.052]	+0.027 [+0.016,+0.052]	+0.024 [+0.012,+0.042]	+0.018 [+0.006,+0.025]	+0.011 [+0.004,+0.016]	+0.006 [+0.002,+0.009]

Note that the confidence intervals for naive and SASC overlap at several budgets even where the paired test is highly significant. This is what pairing predicts: the settings are heterogeneous (tail coverage varies more between settings than between methods), so the unpaired intervals are wide while the per-setting differences are consistent in sign. The paired test is the right one here, since both methods see identical predictions, calibration points and test points by construction (§L).

Every source-using rule against the control. Table 1 in the main text reports the per-draw gain of each rule over the width-matched control at k=20 and k=50. Table 70 gives the underlying tail coverage, mean set size and control value. The picture is the same for every rule: plain pooling and density-ratio pooling, which already fall below naive at the same nominal level for four of the five TFMs (Table 24), stay below the control at k=50, since they return narrower sets than naive and lose coverage with them, and the control buys the same width more efficiently; WCC-CSPD's small gain at k=20 is gone at k=50; and only the two gated rules keep a per-draw gain, NexCP-gated's larger than SASC's, both shrinking with k. The gate, which admits the source only when a held-out target check passes, is what survives the control, not any particular way of using the source. Paired cluster-bootstrap intervals for NexCP-gated, Pooled-DR and WCC-CSPD are in Table 71: NexCP-gated's gain excludes zero for every TFM at $k \leq 100$ under both block structures; Pooled-DR's point estimates are negative for four TFMs with intervals that cross zero; WCC-CSPD's k=20 gain

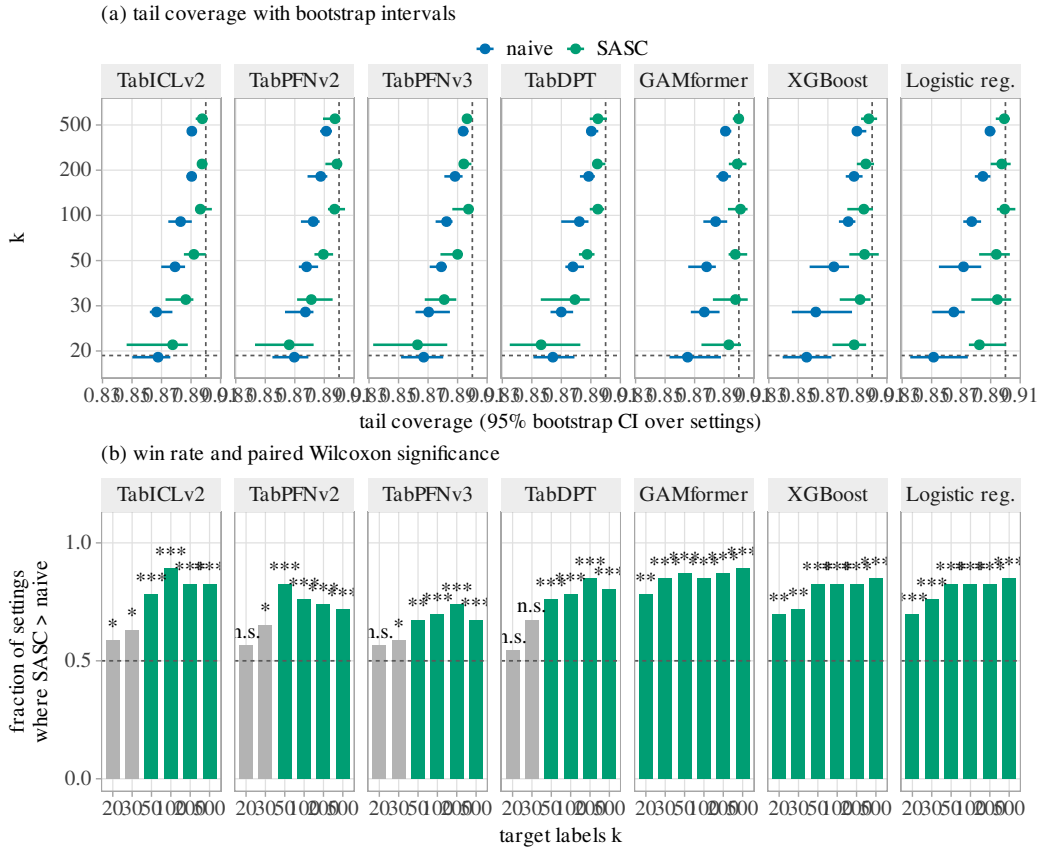


Figure 31: *(a)* Tail coverage with 95% bootstrap intervals over settings (2000 resamples). *(b)* Fraction of settings in which SASC covers better than naive, with paired Wilcoxon significance ($^{***}p < 10^{-4}$, $^{**}p < 0.01$, $^{*}p < 0.05$).

crosses zero except for GAMformer, and its deficit at k=100 excludes zero for four TFMs under setting blocks.

Y Strengthening the method: what worked and what did not

All studies here post-process the cached predictions, with five seeds, both real-shift families and five base models unless stated.

Y.1 Conditional anchoring on model-confidence bins

Confidence bins are the tertiles of $\max_y \hat{p}(y \mid x)$ on the source calibration set; we bin test points by the same thresholds. Within each bin the rule is either the naive target quantile of that bin's labels ("naive, per bin") or SASC anchored to that bin's source scores ("SASC, per bin"). A bin with fewer than eight labels or thirty source scores falls back to the marginal rule. Table 72 reports marginal tail coverage, worst-bin coverage and set size. First, marginal rules have a conditional deficit of about 0.15 on the lowest-confidence bin, and it does not close with more labels: a single threshold cannot serve bins whose score distributions moved by different amounts. Second, per-bin anchoring beats per-bin naive at every budget and for all five base models. The margin is largest at k=50, where each bin holds about seventeen labels, the regime in which Prop. 4 gives the anchor its largest advantage. The price is the set-size cost of any Mondrian rule, about 8% over marginal naive.

Table 69: Bootstrap confidence intervals and paired tests. Tail (10th-percentile) coverage with 95% bootstrap CIs over the 46 settings; win rate is the fraction of settings where SASC $>$ naive; p from a paired Wilcoxon signed-rank test. The last two columns compare SASC with NexCP-gated: the two are indistinguishable at k=50–100.

model	k	naive tail [95% CI]	SASC tail [95% CI]	win rate	Wilcoxon p	NexCP tail	p (SASC vs NexCP)
TabICLv2	20	0.868 [0.850,0.876]	0.878 [0.846,0.888]	0.59	0.0208	0.888	0.149
	30	0.867 [0.862,0.877]	0.886 [0.873,0.892]	0.63	0.0110	0.893	0.160
	50	0.879 [0.870,0.886]	0.892 [0.885,0.900]	0.78	$< 10^{-4}$	0.898	0.857
	100	0.883 [0.875,0.891]	0.896 [0.893,0.904]	0.89	$< 10^{-4}$	0.896	0.054
	200	0.890 [0.887,0.893]	0.897 [0.896,0.901]	0.83	$< 10^{-4}$	0.898	0.041
	500	0.891 [0.889,0.893]	0.898 [0.893,0.901]	0.83	$< 10^{-4}$	0.896	0.008
TabPFNv2	20	0.870 [0.855,0.879]	0.866 [0.843,0.883]	0.57	0.4287	0.891	0.006
	30	0.877 [0.863,0.883]	0.881 [0.872,0.896]	0.65	0.0127	0.894	0.147
	50	0.878 [0.873,0.886]	0.889 [0.883,0.896]	0.83	$< 10^{-4}$	0.897	0.519
	100	0.882 [0.874,0.887]	0.897 [0.892,0.904]	0.76	$< 10^{-4}$	0.897	0.187
	200	0.888 [0.879,0.892]	0.899 [0.891,0.901]	0.74	$< 10^{-4}$	0.897	0.126
	500	0.891 [0.887,0.895]	0.897 [0.889,0.900]	0.72	$< 10^{-4}$	0.896	0.062
TabPFNv3	20	0.867 [0.852,0.880]	0.863 [0.833,0.883]	0.57	0.5962	0.891	0.017
	30	0.870 [0.861,0.885]	0.881 [0.868,0.889]	0.59	0.0322	0.890	0.035
	50	0.879 [0.871,0.882]	0.890 [0.878,0.893]	0.67	0.0001	0.894	0.238
	100	0.883 [0.875,0.887]	0.897 [0.886,0.900]	0.70	$< 10^{-4}$	0.896	0.477
	200	0.888 [0.881,0.893]	0.894 [0.891,0.899]	0.74	$< 10^{-4}$	0.896	0.324
	500	0.894 [0.891,0.896]	0.896 [0.894,0.899]	0.67	$< 10^{-4}$	0.896	0.050
TabDPT	20	0.864 [0.851,0.879]	0.856 [0.835,0.883]	0.54	0.4315	0.892	0.023
	30	0.870 [0.863,0.878]	0.879 [0.856,0.889]	0.67	0.0777	0.886	0.029
	50	0.878 [0.873,0.885]	0.887 [0.882,0.892]	0.76	$< 10^{-4}$	0.895	0.154
	100	0.882 [0.870,0.888]	0.895 [0.889,0.899]	0.78	$< 10^{-4}$	0.891	0.624
	200	0.889 [0.883,0.893]	0.894 [0.892,0.900]	0.85	$< 10^{-4}$	0.894	0.097
	500	0.890 [0.888,0.895]	0.895 [0.889,0.901]	0.80	$< 10^{-4}$	0.895	0.026
GAMformer	20	0.865 [0.853,0.888]	0.893 [0.875,0.901]	0.78	0.0002	0.892	0.531
	30	0.877 [0.867,0.887]	0.898 [0.882,0.906]	0.85	$< 10^{-4}$	0.894	0.114
	50	0.878 [0.866,0.884]	0.898 [0.893,0.906]	0.87	$< 10^{-4}$	0.896	0.010
	100	0.884 [0.876,0.892]	0.901 [0.893,0.906]	0.85	$< 10^{-4}$	0.901	0.011
	200	0.889 [0.885,0.895]	0.899 [0.893,0.905]	0.87	$< 10^{-4}$	0.896	0.001
	500	0.891 [0.888,0.895]	0.900 [0.896,0.903]	0.89	$< 10^{-4}$	0.895	0.000
XGBoost	20	0.856 [0.839,0.872]	0.888 [0.873,0.896]	0.70	0.0005	0.882	0.254
	30	0.862 [0.846,0.886]	0.892 [0.878,0.899]	0.72	0.0001	0.890	0.374
	50	0.874 [0.858,0.884]	0.895 [0.885,0.904]	0.83	$< 10^{-4}$	0.892	0.022
	100	0.884 [0.877,0.889]	0.894 [0.883,0.901]	0.83	$< 10^{-4}$	0.893	0.070
	200	0.888 [0.882,0.894]	0.896 [0.890,0.901]	0.83	$< 10^{-4}$	0.890	0.042
	500	0.890 [0.888,0.896]	0.898 [0.892,0.903]	0.85	$< 10^{-4}$	0.896	0.003
Logistic reg.	20	0.851 [0.836,0.875]	0.882 [0.875,0.900]	0.70	$< 10^{-4}$	0.895	0.211
	30	0.865 [0.850,0.872]	0.895 [0.877,0.904]	0.76	$< 10^{-4}$	0.892	0.800
	50	0.872 [0.855,0.884]	0.894 [0.882,0.903]	0.83	$< 10^{-4}$	0.896	0.541
	100	0.877 [0.871,0.884]	0.899 [0.894,0.907]	0.83	$< 10^{-4}$	0.895	0.088
	200	0.885 [0.879,0.890]	0.898 [0.890,0.904]	0.83	$< 10^{-4}$	0.896	0.156
	500	0.890 [0.887,0.893]	0.899 [0.893,0.903]	0.85	$< 10^{-4}$	0.896	0.021

Table 70: Every source-using rule against the width-matched control, per calibration draw, 46 real-shift settings, 5 seeds, 10 draws each. "tail" is the 10th percentile of coverage over draws, "control" the same statistic for target-only calibration at an inflated rank interpolated at the rule's mean set size, "gain" their difference (bold when positive).

		k=20				k=50			
base model	rule	tail	size	control	gain	tail	size	control	gain
TabICLv2	SASC (gated anchor)	0.857	1.47	0.829	**+0.028**	0.874	1.47	0.863	**+0.011**
	NexCP-gated	0.872	1.47	0.829	**+0.043**	0.879	1.46	0.858	**+0.021**
	Pooled	0.829	1.42	0.820	**+0.009**	0.839	1.41	0.845	-0.006
	Pooled-DR	0.825	1.34	0.820	**+0.005**	0.826	1.34	0.845	-0.018
	WCC-CSPD	0.838	1.41	0.820	**+0.018**	0.840	1.42	0.845	-0.005
TabPFNv2	SASC (gated anchor)	0.851	1.45	0.822	**+0.029**	0.872	1.45	0.857	**+0.015**
	NexCP-gated	0.872	1.46	0.826	**+0.046**	0.878	1.44	0.855	**+0.024**
	Pooled	0.794	1.38	0.820	-0.026	0.785	1.37	0.843	-0.059
	Pooled-DR	0.812	1.33	0.820	-0.008	0.813	1.33	0.843	-0.030
	WCC-CSPD	0.838	1.42	0.820	**+0.018**	0.838	1.42	0.845	-0.007
TabPFNv3	SASC (gated anchor)	0.848	1.44	0.819	**+0.029**	0.869	1.45	0.858	**+0.011**
	NexCP-gated	0.871	1.45	0.824	**+0.047**	0.878	1.44	0.855	**+0.022**
	Pooled	0.782	1.38	0.818	-0.036	0.772	1.36	0.846	-0.074
	Pooled-DR	0.806	1.31	0.818	-0.012	0.810	1.32	0.846	-0.036
	WCC-CSPD	0.833	1.40	0.818	**+0.015**	0.836	1.40	0.846	-0.010
TabDPT	SASC (gated anchor)	0.844	1.44	0.820	**+0.024**	0.871	1.45	0.857	**+0.013**
	NexCP-gated	0.868	1.45	0.825	**+0.043**	0.876	1.44	0.853	**+0.022**
	Pooled	0.747	1.37	0.820	-0.073	0.735	1.35	0.843	-0.108
	Pooled-DR	0.790	1.31	0.820	-0.030	0.794	1.31	0.843	-0.049
	WCC-CSPD	0.828	1.40	0.820	**+0.008**	0.830	1.40	0.843	-0.013
GAMformer	SASC (gated anchor)	0.864	1.53	0.838	**+0.026**	0.883	1.52	0.870	**+0.013**
	NexCP-gated	0.879	1.51	0.833	**+0.047**	0.883	1.50	0.861	**+0.022**
	Pooled	0.850	1.49	0.825	**+0.026**	0.866	1.49	0.856	**+0.010**
	Pooled-DR	0.868	1.43	0.819	**+0.049**	0.868	1.43	0.846	**+0.023**
	WCC-CSPD	0.858	1.48	0.821	**+0.037**	0.858	1.48	0.855	**+0.003**

Table 71: Paired cluster-bootstrap intervals (95%, 1,000 replicates) for the per-draw gain of NexCP-gated, Pooled-DR and WCC-CSPD over the width-matched control, resampling settings or whole tasks. Bold marks intervals that exclude zero.

rule	blocks	base model	k=20	k=50	k=100
NexCP-gated	settings	TabICLv2	**+0.043 [+0.035,+0.054]**	**+0.021 [+0.014,+0.026]**	**+0.011 [+0.004,+0.016]**
		TabPFNv2	**+0.046 [+0.036,+0.056]**	**+0.024 [+0.015,+0.031]**	**+0.012 [+0.006,+0.017]**
		TabPFNv3	**+0.047 [+0.036,+0.058]**	**+0.022 [+0.015,+0.028]**	**+0.011 [+0.006,+0.015]**
		TabDPT	**+0.043 [+0.030,+0.053]**	**+0.022 [+0.017,+0.029]**	**+0.009 [+0.005,+0.017]**
		GAMformer	**+0.047 [+0.039,+0.056]**	**+0.022 [+0.015,+0.029]**	**+0.011 [+0.006,+0.017]**
NexCP-gated	tasks	TabICLv2	**+0.043 [+0.023,+0.057]**	**+0.021 [+0.009,+0.028]**	**+0.011 [+0.000,+0.016]**
		TabPFNv2	**+0.046 [+0.026,+0.058]**	**+0.024 [+0.013,+0.033]**	**+0.012 [+0.004,+0.017]**
		TabPFNv3	**+0.047 [+0.022,+0.065]**	**+0.022 [+0.010,+0.029]**	**+0.011 [+0.003,+0.016]**
		TabDPT	**+0.043 [+0.017,+0.059]**	**+0.022 [+0.014,+0.031]**	**+0.009 [+0.005,+0.019]**
		GAMformer	**+0.047 [+0.037,+0.059]**	**+0.022 [+0.015,+0.032]**	**+0.011 [+0.007,+0.018]**
Pooled-DR	settings	TabICLv2	+0.005 [-0.039,+0.046]	-0.018 [-0.061,+0.020]	-0.029 [-0.072,+0.006]
		TabPFNv2	-0.008 [-0.092,+0.043]	-0.030 [-0.111,+0.021]	-0.044 [-0.124,+0.005]
		TabPFNv3	-0.012 [-0.106,+0.039]	-0.036 [-0.131,+0.012]	-0.047 [-0.137,+0.000]
		TabDPT	-0.030 [-0.084,+0.033]	-0.049 [-0.103,+0.010]	**-0.061 [-0.113,-0.006]**
		GAMformer	**+0.049 [+0.015,+0.064]**	+0.023 [-0.010,+0.035]	+0.010 [-0.024,+0.019]
Pooled-DR	tasks	TabICLv2	+0.005 [-0.149,+0.061]	-0.018 [-0.167,+0.036]	-0.029 [-0.149,+0.020]
		TabPFNv2	-0.008 [-0.175,+0.063]	-0.030 [-0.194,+0.041]	-0.044 [-0.187,+0.021]
		TabPFNv3	-0.012 [-0.180,+0.069]	-0.036 [-0.206,+0.034]	-0.047 [-0.188,+0.021]
		TabDPT	-0.030 [-0.130,+0.063]	-0.049 [-0.151,+0.039]	-0.061 [-0.146,+0.021]
		GAMformer	+0.049 [-0.057,+0.074]	+0.023 [-0.083,+0.045]	+0.010 [-0.082,+0.028]
WCC-CSPD	settings	TabICLv2	+0.018 [-0.021,+0.038]	-0.005 [-0.041,+0.011]	**-0.018 [-0.048,-0.002]**
		TabPFNv2	+0.018 [-0.028,+0.040]	-0.007 [-0.045,+0.014]	**-0.024 [-0.060,-0.001]**
		TabPFNv3	+0.015 [-0.025,+0.037]	-0.010 [-0.048,+0.008]	**-0.022 [-0.058,-0.005]**
		TabDPT	+0.008 [-0.020,+0.033]	-0.013 [-0.039,+0.009]	**-0.026 [-0.050,-0.008]**
		GAMformer	**+0.037 [+0.020,+0.049]**	+0.003 [-0.012,+0.018]	-0.015 [-0.029,+0.000]
WCC-CSPD	tasks	TabICLv2	+0.018 [-0.004,+0.041]	-0.005 [-0.026,+0.016]	-0.018 [-0.041,+0.003]
		TabPFNv2	+0.018 [-0.024,+0.046]	-0.007 [-0.044,+0.024]	-0.024 [-0.058,+0.003]
		TabPFNv3	+0.015 [-0.033,+0.044]	-0.010 [-0.052,+0.008]	**-0.022 [-0.060,-0.005]**
		TabDPT	+0.008 [-0.016,+0.042]	-0.013 [-0.036,+0.015]	**-0.026 [-0.048,-0.002]**
		GAMformer	**+0.037 [+0.010,+0.066]**	+0.003 [-0.019,+0.039]	-0.015 [-0.038,+0.020]

Table 72: Conditional anchoring: marginal tail coverage, worst confidence-bin coverage, and set size at k=50 and k=200, five base models, both families.

family	model	rule	k=50			k=200		
			marg. worst	worst bin	size	marg. worst	worst bin	size
ACS+Wine	TabICLv2	naive	0.879	0.754	1.42	0.890	0.750	1.41
		SASC	0.892	0.800	1.46	0.897	0.784	1.45
		naive, per bin	0.906	0.806	1.57	0.893	0.854	1.49
		SASC, per bin	0.899	0.835	1.54	0.907	0.869	1.52
ACS+Wine	TabPFNv2	naive	0.878	0.760	1.40	0.888	0.753	1.40
		SASC	0.889	0.776	1.44	0.899	0.764	1.43
		naive, per bin	0.906	0.820	1.57	0.894	0.852	1.48
		SASC, per bin	0.906	0.846	1.54	0.908	0.865	1.51
ACS+Wine	TabPFNv3	naive	0.879	0.766	1.40	0.888	0.763	1.39
		SASC	0.890	0.788	1.44	0.894	0.773	1.43
		naive, per bin	0.903	0.810	1.56	0.896	0.850	1.47
		SASC, per bin	0.899	0.836	1.52	0.907	0.863	1.50
ACS+Wine	TabDPT	naive	0.878	0.768	1.40	0.889	0.762	1.39
		SASC	0.887	0.767	1.44	0.894	0.762	1.43
		naive, per bin	0.903	0.819	1.57	0.892	0.855	1.47
		SASC, per bin	0.906	0.843	1.54	0.908	0.875	1.51
ACS+Wine	GAMformer	naive	0.878	0.711	1.44	0.889	0.712	1.44
		SASC	0.898	0.795	1.52	0.899	0.786	1.51
		naive, per bin	0.905	0.835	1.60	0.893	0.859	1.52
		SASC, per bin	0.897	0.833	1.58	0.910	0.870	1.56
ACS+Wine	XGBoost	naive	0.874	0.719	1.45	0.888	0.719	1.45
		SASC	0.895	0.778	1.51	0.896	0.753	1.49
		naive, per bin	0.901	0.812	1.60	0.895	0.857	1.52
		SASC, per bin	0.906	0.845	1.57	0.905	0.876	1.55
ACS+Wine	Logistic reg.	naive	0.872	0.729	1.48	0.885	0.707	1.47
		SASC	0.894	0.767	1.53	0.898	0.742	1.52
		naive, per bin	0.901	0.817	1.63	0.890	0.852	1.53
		SASC, per bin	0.899	0.845	1.60	0.905	0.874	1.58
UCI	TabICLv2	naive	0.887	0.755	1.55	0.895	0.755	1.54
		SASC	0.895	0.769	1.69	0.900	0.762	1.59
		naive, per bin	0.893	0.798	1.65	0.888	0.833	1.58
		SASC, per bin	0.898	0.821	1.68	0.895	0.873	1.60
UCI	TabPFNv2	naive	0.881	0.725	1.64	0.889	0.684	1.63
		SASC	0.880	0.771	1.73	0.893	0.784	1.73
		naive, per bin	0.880	0.790	1.74	0.881	0.782	1.66
		SASC, per bin	0.898	0.812	1.76	0.900	0.860	1.69
UCI	TabPFNv3	naive	0.880	0.746	1.56	0.889	0.746	1.54
		SASC	0.897	0.810	1.65	0.890	0.795	1.65
		naive, per bin	0.897	0.826	1.66	0.887	0.842	1.58
		SASC, per bin	0.894	0.839	1.67	0.891	0.861	1.61
UCI	TabDPT	naive	0.887	0.727	1.65	0.888	0.752	1.61
		SASC	0.896	0.775	1.73	0.888	0.777	1.69
		naive, per bin	0.894	0.844	1.77	0.888	0.864	1.65
		SASC, per bin	0.903	0.848	1.78	0.888	0.861	1.70
UCI	GAMformer	naive	0.889	0.779	2.34	0.890	0.769	2.20
		SASC	0.890	0.783	2.61	0.898	0.790	2.47
		naive, per bin	0.913	0.846	2.32	0.893	0.844	2.24
		SASC, per bin	0.908	0.831	2.51	0.900	0.871	2.50
UCI	XGBoost	naive	0.895	0.690	1.97	0.893	0.672	1.90
		SASC	0.899	0.812	2.21	0.894	0.811	2.15
		naive, per bin	0.905	0.763	2.10	0.896	0.860	1.97
		SASC, per bin	0.896	0.777	2.17	0.902	0.865	2.04
UCI	Logistic reg.	naive	0.871	0.659	1.72	0.891	0.693	1.71
		SASC	0.893	0.757	1.76	0.895	0.768	1.74
		naive, per bin	0.872	0.716	1.89	0.895	0.858	1.85
		SASC, per bin	0.892	0.822	1.86	0.901	0.845	1.84

Y.2 TARGETED LABELING

Targeted labeling chooses *which* target points to label. We draw the k points with probability $\pi(x) \propto 0.15 + \exp\{-(s_{\text{top}}(x) - q_S)^2/2\sigma^2\}$, where $s_{\text{top}} = 1 - \max_y \hat{p}(y \mid x)$ and q_S is the source quantile, so labels concentrate near the quantile that decides coverage. The design is known, so weighted conformal with weights $1/\pi$ is exactly valid (Tibshirani et al., 2019).

We draw the k labeled points from the 500-point target pool with probability proportional to a design $\pi(x)$: uniform; "low-confidence", $\pi \propto 0.15 + s_{\text{top}}(x)$; or "quantile band", $\pi \propto 0.15 + \exp\{-(s_{\text{top}}(x) - q_S)^2/(2 \cdot 0.1^2)\}$. Here s_{top} is the score under the model's top label and q_S the source quantile, both available before labeling. Calibration is weighted conformal with weights $1/\pi$ on the labeled points and on the test point (Tibshirani et al., 2019), which is exactly valid because the design is known. SASC uses the weighted median and quantile in place of the unweighted ones. We average each (setting, seed, k, design) over ten independent draws, so the numbers in Table 73 are comparable within the table but smoother than those of §L. Both non-uniform designs raise tail coverage at every budget; the quantile band is best from k=50 on, and the gain is largest at k=20 (+0.016 to +0.021). Adding the anchor on top of targeted labels changes coverage by at most 0.002: the two attack the same variance and do not compound.

Table 73: Targeted labeling: tail coverage for three sampling designs, with and without the source anchor, both TFMs and both families.

family	rule	design	k=20	k=50	k=100	k=200	size$_{k=50}$
ACS+Wine, TabICLv2							
	naive	random labels	0.893	0.893	0.894	0.893	1.42
	naive	low-confidence design	0.914	0.897	0.895	0.896	1.45
	naive	quantile-band design	0.909	0.902	0.899	0.895	1.47
	SASC	random labels	0.888	0.895	0.897	0.897	1.45
	SASC	low-confidence design	0.902	0.898	0.898	0.897	1.46
	SASC	quantile-band design	0.910	0.903	0.898	0.898	1.48
ACS+Wine, TabPFNv2							
	naive	random labels	0.892	0.894	0.892	0.892	1.42
	naive	low-confidence design	0.909	0.896	0.896	0.893	1.44
	naive	quantile-band design	0.913	0.906	0.900	0.894	1.46
	SASC	random labels	0.875	0.889	0.892	0.893	1.43
	SASC	low-confidence design	0.894	0.895	0.896	0.894	1.45
	SASC	quantile-band design	0.886	0.898	0.898	0.897	1.46
UCI family, TabICLv2							
	naive	random labels	0.890	0.897	0.896	0.898	1.56
	naive	low-confidence design	0.917	0.902	0.894	0.887	1.61
	naive	quantile-band design	0.925	0.909	0.903	0.899	1.65
	SASC	random labels	0.840	0.878	0.883	0.893	1.62
	SASC	low-confidence design	0.875	0.894	0.890	0.888	1.65
	SASC	quantile-band design	0.835	0.878	0.887	0.893	1.62
UCI family, TabPFNv2							
	naive	random labels	0.887	0.895	0.893	0.895	1.65
	naive	low-confidence design	0.902	0.898	0.896	0.883	1.71
	naive	quantile-band design	0.911	0.901	0.894	0.893	1.71
	SASC	random labels	0.840	0.876	0.881	0.890	1.72
	SASC	low-confidence design	0.876	0.885	0.891	0.886	1.77
	SASC	quantile-band design	0.872	0.878	0.884	0.888	1.72

Sensitivity to the binning. Table 74 varies the number of confidence bins (2, 3, 5) and the fallback threshold (5, 8, 15 labels). Per-bin anchoring is at or above per-bin naive in 97 of the 108 cells. The eleven exceptions are within 0.02, and all but one occur with two bins, where per-bin naive already has enough labels. The margin grows as bins become more label-starved: with three bins and a strict fallback of 15 labels it is 0.824 against 0.767 at k=50 for TabICL. Worst-bin coverage falls with more bins for both rules, since the metric itself gets harder.

Table 74: Conditional anchoring under different numbers of bins and fallback thresholds: worst-bin coverage, 10th percentile over settings.

family	model	bins	fallback	k=50		k=100		k=200	
				naive/bin	SASC/bin	naive/bin	SASC/bin	naive/bin	SASC/bin
ACS+Wine	TabICLv2	2	5	0.854	0.850	0.868	0.866	0.869	0.882
ACS+Wine	TabICLv2	2	8	0.840	0.846	0.868	0.866	0.869	0.882
ACS+Wine	TabICLv2	2	15	0.840	0.850	0.852	0.869	0.869	0.882
ACS+Wine	TabICLv2	3	5	0.818	0.835	0.847	0.854	0.854	0.869
ACS+Wine	TabICLv2	3	8	0.806	0.835	0.846	0.854	0.854	0.869
ACS+Wine	TabICLv2	3	15	0.767	0.824	0.824	0.852	0.854	0.869
ACS+Wine	TabICLv2	5	5	0.773	0.782	0.801	0.822	0.835	0.844
ACS+Wine	TabICLv2	5	8	0.738	0.764	0.788	0.820	0.834	0.844
ACS+Wine	TabICLv2	5	15	0.703	0.741	0.760	0.815	0.816	0.851
ACS+Wine	TabPFNv2	2	5	0.859	0.855	0.858	0.874	0.870	0.881
ACS+Wine	TabPFNv2	2	8	0.859	0.855	0.858	0.874	0.870	0.881
ACS+Wine	TabPFNv2	2	15	0.857	0.855	0.858	0.874	0.870	0.881
ACS+Wine	TabPFNv2	3	5	0.827	0.849	0.838	0.854	0.852	0.865
ACS+Wine	TabPFNv2	3	8	0.820	0.846	0.838	0.854	0.852	0.865
ACS+Wine	TabPFNv2	3	15	0.800	0.828	0.838	0.853	0.852	0.865
ACS+Wine	TabPFNv2	5	5	0.779	0.798	0.806	0.819	0.828	0.835
ACS+Wine	TabPFNv2	5	8	0.764	0.783	0.802	0.817	0.826	0.835
ACS+Wine	TabPFNv2	5	15	0.724	0.738	0.780	0.814	0.826	0.831
UCI	TabICLv2	2	5	0.831	0.867	0.864	0.873	0.885	0.882
UCI	TabICLv2	2	8	0.789	0.844	0.864	0.826	0.885	0.882
UCI	TabICLv2	2	15	0.779	0.844	0.765	0.806	0.885	0.882
UCI	TabICLv2	3	5	0.798	0.846	0.838	0.844	0.870	0.873
UCI	TabICLv2	3	8	0.798	0.821	0.796	0.826	0.833	0.873
UCI	TabICLv2	3	15	0.762	0.777	0.762	0.795	0.791	0.867
UCI	TabICLv2	5	5	0.749	0.793	0.752	0.794	0.821	0.847
UCI	TabICLv2	5	8	0.736	0.764	0.752	0.780	0.803	0.846
UCI	TabICLv2	5	15	0.691	0.747	0.738	0.744	0.765	0.828
UCI	TabPFNv2	2	5	0.869	0.878	0.862	0.879	0.873	0.887
UCI	TabPFNv2	2	8	0.767	0.822	0.862	0.879	0.873	0.887
UCI	TabPFNv2	2	15	0.767	0.822	0.862	0.877	0.859	0.887
UCI	TabPFNv2	3	5	0.790	0.812	0.737	0.856	0.850	0.867
UCI	TabPFNv2	3	8	0.790	0.812	0.733	0.837	0.782	0.860
UCI	TabPFNv2	3	15	0.752	0.808	0.733	0.824	0.732	0.842
UCI	TabPFNv2	5	5	0.734	0.778	0.808	0.821	0.822	0.813
UCI	TabPFNv2	5	8	0.695	0.762	0.745	0.796	0.821	0.812
UCI	TabPFNv2	5	15	0.644	0.741	0.691	0.793	0.698	0.813

One draw per setting. The numbers above average ten label draws per setting and seed. Table 75 recomputes them with a single draw, the statistic of the main suite. The gains are unchanged in kind and somewhat larger in size. With a single draw the anchor and the design add partially: SASC on quantile-band labels reaches 0.906 at k=50 for TabICL, against 0.892 for naive on the same labels.

Y.3 A CERTIFIED GATE

The point form's gate compares a held-out coverage estimate with $1-\alpha$. We replace this comparison by a fixed-sequence test in the manner of learn-then-test (Angelopoulos et al., 2025): we scan the weight grid from most to least anchor use and accept the first weight whose Clopper–Pearson lower bound (level $\delta = 0.1$) on held-out coverage reaches $1 - \alpha - \epsilon$. Whenever some weight is accepted, the selection has a training-conditional guarantee at tolerance ϵ, up to treating five-fold held-out indicators as a binomial sample. Table 77 reports the point rule with this gate and a certified rule that falls back to the target-only order statistic when no weight passes. At $\epsilon = 0.02$ the gate certifies a weight in 16–29% of runs at k=50 and in 84–88% at k=500. Because five-fold indicators are not an exact binomial sample, Table 76 repeats the test with a genuine split of the k labels into fitting and testing halves, which makes the Clopper–Pearson bound exact. The certification rate is then 32% at

Table 75: Targeted labeling with one draw per setting and seed, tail coverage over the 46 settings.

model	rule	design	k=20	k=50	k=100	k=200
TabICLv2	naive	random labels	0.877	0.880	0.886	0.888
TabICLv2	naive	low-confidence design	0.897	0.881	0.882	0.888
TabICLv2	naive	quantile-band design	0.896	0.892	0.896	0.899
TabICLv2	SASC	random labels	0.872	0.886	0.894	0.895
TabICLv2	SASC	low-confidence design	0.881	0.893	0.893	0.895
TabICLv2	SASC	quantile-band design	0.888	0.906	0.898	0.899
TabPFNv2	naive	random labels	0.873	0.878	0.887	0.887
TabPFNv2	naive	low-confidence design	0.891	0.887	0.893	0.889
TabPFNv2	naive	quantile-band design	0.881	0.891	0.895	0.896
TabPFNv2	SASC	random labels	0.869	0.883	0.889	0.892
TabPFNv2	SASC	low-confidence design	0.882	0.887	0.892	0.892
TabPFNv2	SASC	quantile-band design	0.884	0.893	0.894	0.898

k=50 and 63% at k=500 on ACS, and the certified rule is never tighter than certified naive at the same coverage. The result is negative: certifying the anchor adds nothing over certifying the target quantile alone.

Table 76: Certified gating with a genuine fit/test split (TabICLv2): tail coverage / set size and the fraction of runs in which a weight is certified.

family	rule	k=50		k=100		k=200		k=500	
		worst	size	worst	size	worst	size	worst	size
ACS+Wine	naive	0.879	1.42	0.883	1.41	0.890	1.41	0.891	1.41
	SASC (CV gate)	0.892	1.46	0.896	1.46	0.897	1.45	0.898	1.45
	SASC (split test, ϵ=0.02)	0.886	1.49	0.880	1.44	0.886	1.44	0.889	1.44
	certified naive	0.943	1.65	0.925	1.56	0.915	1.49	0.912	1.46
	SASC-cons	0.945	1.66	0.935	1.57	0.922	1.51	0.914	1.49
	certified (split, ϵ=0.02)	0.942	1.64	0.929	1.57	0.916	1.50	0.904	1.47
	certification rate	0.32		0.31		0.49		0.63	
UCI	naive	0.887	1.55	0.892	1.54	0.895	1.54	0.895	1.54
	SASC (CV gate)	0.895	1.69	0.898	1.64	0.900	1.59	0.895	1.61
	SASC (split test, ϵ=0.02)	0.899	1.76	0.882	1.67	0.891	1.63	0.890	1.58
	certified naive	0.950	1.84	0.928	1.69	0.915	1.63	0.914	1.60
	SASC-cons	0.950	1.93	0.930	1.77	0.915	1.66	0.914	1.66
	certified (split, ϵ=0.02)	0.950	1.96	0.920	1.78	0.915	1.70	0.908	1.61
	certification rate	0.35		0.40		0.40		0.56	

Y.4 WCC-CSPD WITH PER-CLASS DENSITY-RATIO ESTIMATORS

The main suite instantiates WCC-CSPD with the marginal density ratio times a class-prior ratio. To follow the original construction more closely, we fit a separate balanced logistic domain classifier per class, on the source class-j covariates against the labeled target class-j covariates, and fall back to the marginal weight when a class has fewer than five target labels. This requires regenerating the covariates of every cached split. Table 78 compares the two instantiations. The faithful version is no better on ACS (0.844 against 0.849 at k=50 for TabICL) and worse at k=20, where the per-class classifiers overfit. On the UCI family it is better (0.61 against 0.48) but still far below naive (0.89). The main text reports the faithful numbers.

Y.5 THREE VARIANTS THAT DID NOT HELP

Plug-in weight. We replace the CV gate by the risk-optimal weight of Eq. (26), with densities from the source and a bias-corrected estimate of b^2 from the target labels. This gives mean coverage of exactly 0.900 and tighter sets but no tail gain, and it hurts on the UCI family, where the source density no longer describes the target (Table 79). *Semi-supervised anchor.* Estimating the target median from

Table 77: Certified gating (TabICLv2): tail coverage / set size, and the fraction of runs in which a weight is certified.

family	rule	k=50		k=100		k=200		k=500	
		worst	size	worst	size	worst	size	worst	size
ACS+Wine	naive	0.879	1.42	0.883	1.41	0.890	1.41	0.891	1.41
	SASC (CV gate)	0.892	1.46	0.896	1.46	0.897	1.45	0.898	1.45
	SASC (LTT gate, ϵ=0.02)	0.889	1.46	0.895	1.46	0.899	1.45	0.898	1.45
	certified naive	0.943	1.65	0.925	1.56	0.915	1.49	0.912	1.46
	SASC-cons	0.945	1.66	0.935	1.57	0.922	1.51	0.914	1.49
	certified LTT (ϵ=0.02)	0.941	1.65	0.931	1.57	0.914	1.51	0.899	1.46
	certification rate	0.20		0.30		0.39		0.88	
UCI	naive	0.887	1.55	0.892	1.54	0.895	1.54	0.895	1.54
	SASC (CV gate)	0.895	1.69	0.898	1.64	0.900	1.59	0.895	1.61
	SASC (LTT gate, ϵ=0.02)	0.893	1.65	0.898	1.64	0.900	1.59	0.895	1.61
	certified naive	0.950	1.84	0.928	1.69	0.915	1.63	0.914	1.60
	SASC-cons	0.950	1.93	0.930	1.77	0.915	1.66	0.914	1.66
	certified LTT (ϵ=0.02)	0.950	1.90	0.928	1.77	0.915	1.66	0.895	1.62
	certification rate	0.24		0.25		0.25		0.87	

Table 78: Two instantiations of WCC-CSPD: tail coverage against budget.

family	model	rule	k=20	k=50	k=100	k=200	k=500	size$_{k=50}$
ACS+Wine	TabICLv2	naive	0.868	0.879	0.883	0.890	0.891	1.42
ACS+Wine	TabICLv2	WCC-CSPD (marginal-ratio approx.)	0.847	0.849	0.851	0.855	0.861	1.41
ACS+Wine	TabICLv2	WCC-CSPD (per-class domain classifier)	0.802	0.844	0.860	0.864	0.867	1.39
ACS+Wine	TabICLv2	SASC	0.878	0.892	0.896	0.897	0.898	1.46
ACS+Wine	TabPFNv2	naive	0.870	0.878	0.882	0.888	0.891	1.40
ACS+Wine	TabPFNv2	WCC-CSPD (marginal-ratio approx.)	0.846	0.848	0.849	0.852	0.861	1.42
ACS+Wine	TabPFNv2	WCC-CSPD (per-class domain classifier)	0.802	0.841	0.848	0.856	0.866	1.39
ACS+Wine	TabPFNv2	SASC	0.866	0.889	0.897	0.899	0.897	1.44
UCI	TabICLv2	naive	0.892	0.887	0.892	0.895	0.895	1.55
UCI	TabICLv2	WCC-CSPD (marginal-ratio approx.)	0.475	0.478	0.505	0.581	0.690	1.38
UCI	TabICLv2	WCC-CSPD (per-class domain classifier)	0.410	0.406	0.499	0.571	0.686	1.41
UCI	TabICLv2	SASC	0.854	0.895	0.898	0.900	0.895	1.69
UCI	TabPFNv2	naive	0.875	0.881	0.887	0.889	0.896	1.64
UCI	TabPFNv2	WCC-CSPD (marginal-ratio approx.)	0.418	0.435	0.457	0.569	0.666	1.36
UCI	TabPFNv2	WCC-CSPD (per-class domain classifier)	0.365	0.376	0.440	0.542	0.689	1.39
UCI	TabPFNv2	SASC	0.873	0.880	0.892	0.893	0.896	1.73

all 500 pool points under the model's own labels, corrected by the labeled points, matches the labeled median and changes nothing. *Multi-source anchor.* Anchoring to the state whose score distribution is closest to the k target scores, or to a W_1-weighted mixture of all nine other states, is no better than anchoring to the training source (Table 80). The closest state is chosen noisily from k points, and the training source already has the smallest bias for most targets.

Y.6 Negative results from the main text in detail

§4.3 summarizes three ideas that do not work. We give the full account here, because a negative result is only useful if the reader can see the idea got a fair chance.

(i) Embedding-space density ratios. The motivation is strong: TabICL computes a 512-dimensional row embedding, its own learned notion of similarity, so estimating $\mathrm{d}P_T/\mathrm{d}P_S$ there rather than in raw feature space should give better weights. Our first attempt failed for a mundane reason: a logistic domain classifier in 512 dimensions with $n < d$ overfits and produces degenerate weights. We therefore ran a fair rematch with PCA to 16 dimensions, stronger ℓ_2 regularization ($C = 0.1$), and a finer temper grid at the low end. The embedding still lost, with a coverage/size frontier of 0.923/1.549 against feature space's 0.903/1.476. Our diagnosis is that a representation trained to *separate* classes also separates calibration from test more sharply than raw features do,

Table 79: Variants that did not help: tail coverage against SASC with the CV gate.

family	model	variant	variant				SASC (CV gate)			
			k=20	k=50	k=100	k=500k=20	k=50	k=100	k=500	
ACS+Wine	TabICLv2	plug-in $w^\star$	0.866	0.878	0.881	0.888	0.878	0.892	0.896	0.898
ACS+Wine	TabPFNv2	plug-in $w^\star$	0.867	0.877	0.885	0.889	0.866	0.889	0.897	0.897
ACS+Wine	TabICLv2	plug-in $w^\star$, gated	0.874	0.883	0.889	0.894	0.878	0.892	0.896	0.898
ACS+Wine	TabPFNv2	plug-in $w^\star$, gated	0.869	0.884	0.888	0.891	0.866	0.889	0.897	0.897
UCI	TabICLv2	plug-in $w^\star$	0.818	0.839	0.858	0.884	0.854	0.895	0.898	0.895
UCI	TabPFNv2	plug-in $w^\star$	0.836	0.842	0.855	0.885	0.873	0.880	0.892	0.896
UCI	TabICLv2	plug-in $w^\star$, gated	0.877	0.895	0.898	0.895	0.854	0.895	0.898	0.895
UCI	TabPFNv2	plug-in $w^\star$, gated	0.876	0.878	0.886	0.896	0.873	0.880	0.892	0.896
ACS+Wine	TabICLv2	semi-supervised anchor	0.876	0.894	0.894	0.898	0.878	0.892	0.896	0.898
ACS+Wine	TabPFNv2	semi-supervised anchor	0.870	0.889	0.895	0.897	0.866	0.889	0.897	0.897
UCI	TabICLv2	semi-supervised anchor	0.854	0.896	0.898	0.895	0.854	0.895	0.898	0.895
UCI	TabPFNv2	semi-supervised anchor	0.879	0.880	0.892	0.896	0.873	0.880	0.892	0.896

Table 80: Multi-source anchors on ACS: tail coverage and set size.

model	anchor	k=20	k=50	k=100	k=200	k=500	size$_{k=50}$
TabICLv2	naive	0.868	0.879	0.883	0.890	0.890	1.41
	SASC, California anchor	0.877	0.892	0.896	0.897	0.897	1.45
	SASC, closest-state anchor	0.874	0.890	0.893	0.896	0.896	1.44
	SASC, W_1-weighted mixture anchor	0.864	0.890	0.891	0.894	0.893	1.44
TabPFNv2	naive	0.869	0.878	0.882	0.887	0.891	1.39
	SASC, California anchor	0.866	0.889	0.897	0.898	0.897	1.43
	SASC, closest-state anchor	0.876	0.889	0.895	0.893	0.895	1.42
	SASC, W_1-weighted mixture anchor	0.872	0.887	0.888	0.885	0.890	1.42

which produces extreme weights and collapses n_{eff}. That is the opposite of what weight estimation needs: quality for prediction and suitability for density-ratio estimation are different properties of a representation, and here they conflict.

(ii) Support-set curation. In-context learning allows an intervention ordinary models do not: resample the support set toward the target covariate distribution, so the model conditions on data like its deployment population. This is the ICL analogue of what weighted conformal does to the calibration set. It does not help: accuracy is unchanged and ECE moves from 0.070 to 0.075, i.e. slightly worse. Our explanation is that ICL derives its calibration from a *varied* support set; resampling toward the target trades that variety for a covariate alignment the predictor does not exploit. We regard this as the most interesting of the three negatives, because it suggests that support-set interventions should target the *conditional* (which we cannot observe without target labels) rather than the marginal.

(iii) Post-hoc probability calibration. Temperature scaling (Guo et al., 2017) is the standard fix for miscalibrated probabilities, and the TFMs are indeed miscalibrated under shift (ECE rises from 0.035 to 0.10). Fitting a temperature on target data improves mean target ECE from 0.046 to 0.043 and leaves conformal coverage *exactly* unchanged, $0.889 \rightarrow 0.889$, with zero change to machine precision on every one of the 25 task/target pairs (Figure 32). The reason is structural, not empirical: temperature scaling is a monotone transformation of the scores, and the conformal threshold is an order statistic of those same scores, so the transformation carries the threshold along and the induced prediction sets are identical. The ECE improvement is also not uniform: temperature scaling makes calibration *worse* on 10 of the 25 pairs, so even on its own terms it is not a free win under shift. We say this explicitly because the intuitive response to our diagnosis is to recalibrate the miscalibrated model. That confuses two things: calibration concerns the *values* of predicted probabilities, while coverage concerns the *ranking* of scores relative to a quantile of the right distribution, and only target labels move the latter.

What the three failures have in common. Each tries to fix a concept-shift problem with a covariate-side or monotone-transformation tool: better weights (i), better support covariates (ii), better probability values (iii). None of them can change the effect of $P_T(Y \mid X)$ on the score distribution. By contrast, the one elaborate method that *does* help, class-conditional calibration, models

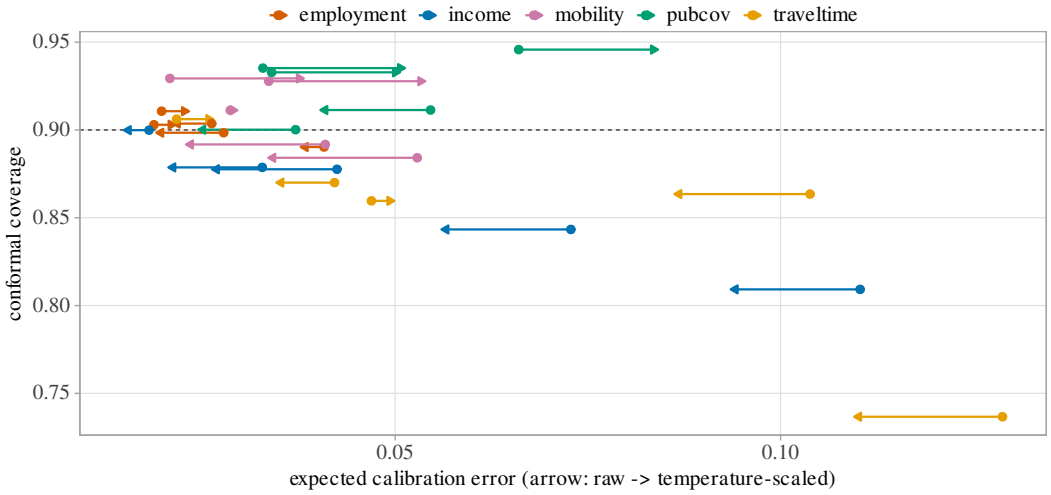


Figure 32: Each arrow is one ACS task/target pair, from the raw model to the temperature-scaled one. Every arrow is horizontal, since temperature scaling is monotone and conformal coverage is unchanged for all 25 pairs, while ECE moves both ways. Dashed: nominal 0.90.

the per-class score shift that concept shift creates, so under concept shift added machinery pays only when it targets the conditional.

Z Broader impact and ethical considerations

This work aims to make the uncertainty estimates of tabular models more trustworthy under distribution shift. Its most likely effect is to make deployed models more reliable about what they do not know. The application domains we use as motivation are credit, healthcare eligibility and public-benefit coverage. There an over-confident prediction set does real harm, and the failure mode we document, coverage silently dropping from 90% to 70% when a model moves to a new state or year, would otherwise be invisible to the operator.

Three caveats belong on the record. First, our method needs *labeled target data*, and collecting labels is not ethically neutral. In some settings the labels are outcomes for real people. Collecting a few hundred means either waiting for outcomes to arrive or running the model on a population before it is validated for them. Practitioners should treat the labeling budget as part of the deployment plan, not an afterthought. Second, valid marginal coverage is not fairness: a prediction set can reach 90% coverage overall while covering a protected subgroup far less often. Our class-conditional results (§4.2, §W) show that the machinery to enforce group-wise validity exists and is cheap. We would recommend it with sensitive attributes as the grouping variable, but we have not evaluated that here and do not claim it. Third, the ACS data we use is public and de-identified, and the folktables tasks were built (Ding et al., 2021) to support this kind of methodological work. We use it as a shift benchmark and make no substantive claims about income, employment or coverage in any state.

Finally, a note on what the paper does *not* license. Our conservative variant carries a real training-conditional guarantee; the point form does not, and we say so throughout (§D.4). In a high-stakes deployment that requires a certificate, the conservative form, or plain target-only conformal, is the correct choice, at the price of wider sets. Deploying the efficient heuristic where a guarantee is contractually or ethically required would be a misuse of this work.