

Article

StaCOR: Decision-Stable Output Refinement for TabPFN with Noisy Support Labels

Yihong Huang, Mingjun Shi , Lingkai Kong and Kun She *

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610031, China; 202452090808@std.uestc.edu.cn (Y.H.); averagejoe@std.uestc.edu.cn (M.S.); 202212090807@std.uestc.edu.cn (L.K.)

* Correspondence: kun@uestc.edu.cn

Abstract

TabPFN-style in-context prediction can be degraded by corrupted support labels. When the deployed support context is cached, version-locked, shared, or otherwise operationally constrained from refresh, external adaptation pipelines can still use newly available clean evidence without modifying that served context. We propose StaCOR (Stability-aware Coupling-Regularized Output Refinement), a selectively constrained fixed-context refinement method for frozen TabPFN. StaCOR leaves the noisy support context unchanged and trains a lightweight output-side correction head using auxiliary clean evidence. Regularized multiclass coupling controls correction capacity at the parameter level, while Selective Decision Stability (SDS) applies a training-time zero-margin penalty only to clean examples that the noisy-support baseline predicts correctly; baseline errors remain unconstrained by this stability term. We evaluate seven tabular datasets, including four development benchmarks and three frozen external-validation datasets under symmetric and instance-dependent support-label noise using probabilistic and decision metrics, paired statistical analysis, direct-context alternatives, and serving measurements. Relative to a regularized affine calibration baseline, StaCOR has a statistically indistinguishable benchmark negative log-likelihood (NLL) under symmetric noise and incurs a small NLL cost under instance-dependent noise, while reducing harmful-flip rates from 0.0195 to 0.0124 and from 0.0202 to 0.0124, respectively. External-validation results show the same decision-stability trend, although the calibration baseline can retain higher Macro-F1. SDS ablation further shows more consistent decision preservation, with small measurable NLL costs in some settings. When context modification is feasible, directly using clean examples generally yields lower NLL and higher Macro-F1. StaCOR therefore targets fixed-context or refresh-constrained deployments where a more favorable probability–decision trade-off is desired without modifying the original served support context.



Academic Editors: Juan Ernesto Solanes Galbis and Chang Tang

Received: 27 August 2026

Revised: 28 September 2026

Accepted: 11 October 2026

Published: 16 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: TabPFN; tabular foundation models; label noise; posterior refinement; decision stability; probability calibration; in-context learning

1. Introduction

Tabular prediction remains a central problem in scientific, engineering, and operational machine learning. In many applications, the data are structured as heterogeneous feature columns rather than images, text, or sequences, and users need models that can adapt quickly to new prediction tasks without expensive task-specific retraining. TabPFN-style tabular foundation models address this need by performing in-context prediction: a support

set of labeled examples is provided at inference time, and the model returns a predictive posterior for query examples conditioned on that support context [1,2]. This design makes the support examples an active part of the predictive computation rather than merely a source of offline training data.

The same in-context property also makes support-label quality important. In practical use, support examples may be assembled from historical records, weak labels, partially audited data, or labels that have changed across data versions. Some support labels can therefore be incorrect. Noisy support labels may degrade the predictive posterior, distort class probabilities, and alter downstream classifications. This paper studies that issue in controlled label-noise settings, focusing on symmetric noise (SN) and instance-dependent noise (IDN) as repeatable perturbation protocols rather than as complete models of all real annotation errors [3–5].

One natural response is to leave the TabPFN backbone and its original served support context fixed during an evaluated deployment cycle, then refine the output posterior using a lightweight correction model trained on auxiliary clean evidence. This is an operational rather than informational assumption: clean evidence can be available to an external adaptation pipeline even when the served context is cached, version-locked, shared across a serving system, audited, or costly to rebuild and revalidate. It separates two deployment choices that are often conflated: changing the examples seen by the in-context predictor versus correcting the posterior after the frozen predictor has run. When context modification is operationally acceptable, direct clean-context utilization is the more direct intervention. Output-side calibration methods, including temperature-style rescaling, Dirichlet-style transformations, and last-layer correction models, provide useful reference points for the fixed-context intervention [6–8].

However, improving probability quality is not the same objective as preserving hard classification decisions. Negative log-likelihood (NLL) evaluates the probability assigned to the clean label over the full predictive distribution, while a deployed classifier typically acts on the argmax decision. A correction can redistribute probability mass in a way that improves probabilistic fit while moving a low-margin, originally correct prediction across the decision boundary. Conversely, a correction that avoids such transitions may accept a small probability cost. This distinction motivates examining posterior refinement and decision preservation jointly, especially when the noisy-support baseline already makes some useful decisions that should not be disturbed without evidence [9–11]. The central question is therefore not whether posterior refinement may change decisions, but which decisions should be selectively protected from change.

The structure of the correction model also matters. A purely diagonal transformation can rescale classes independently, but it cannot represent interactions in which evidence for one class should affect another class's corrected logit. A fully flexible cross-class transformation can represent richer adjustments, but the additional degrees of freedom may also permit unnecessary cross-class changes. In the noisy-support setting, this motivates a capacity-control question: the correction should be expressive enough to repair systematic posterior distortions, while restrained enough to avoid avoidable changes to decisions that the frozen baseline already handled correctly. This motivates regularizing, rather than eliminating, cross-class coupling.

This paper introduces StaCOR (Stability-aware Coupling-Regularized Output Refinement), a selectively constrained fixed-context output refinement method for TabPFN under noisy support labels. StaCOR leaves the TabPFN model and the noisy support set unchanged, uses the frozen TabPFN log posterior as input, and trains only a lightweight correction head on clean head-training evidence. The method combines parameter-level control of multiclass posterior correction with sample-level selective protection of baseline-

correct decisions. Regularized Class Coupling (RCC) controls cross-class correction capacity by retaining off-diagonal correction capacity while shrinking cross-class interactions. Selective Decision Stability (SDS) provides a zero-margin boundary-violation penalty for clean training examples that the frozen noisy-support baseline already predicts correctly. Baseline-wrong examples are not constrained by this stability term, so they retain the possibility of beneficial decision changes under the full training objective. This stability term is not a harmful-flip detector and does not predict future query errors; it is a reactive training penalty defined from baseline-correct clean examples.

We evaluate StaCOR on four development benchmarks—Yeast, Wine, PageBlocks, and Optdigits—and on three frozen external-validation datasets—Gas Sensor Drift, Crop Mapping, and Landsat Satellite—under SN and IDN protocols. The comparisons include the noisy TabPFN baseline, temperature scaling, Dirichlet calibration (DC), Variational Bayesian Last Layers (VBLL)-inspired VBLL-D mean and VBLL-full Monte Carlo (MC), the Selective Decision Stability ablation, direct clean-label context policies, and serving-efficiency measurements. The results support StaCOR as offering a competitive probability–decision trade-off rather than as a universal NLL optimizer. In particular, DC retains a small benchmark IDN NLL advantage and higher external Macro-F1, whereas StaCOR provides lower harmful-flip rates and higher retention relative to DC. The ablation indicates that Selective Decision Stability improves the measured decision metrics while carrying small NLL costs in some settings. Direct context utilization generally remains preferable for NLL and Macro-F1 when clean samples can be inserted into or used to replace context examples, so StaCOR is targeted primarily to fixed-context or refresh-constrained deployments.

The main contributions of this work are as follows:

1. We frame noisy-support TabPFN refinement as a selectively constrained fixed-context problem, in which auxiliary clean evidence is used to improve posterior quality without unnecessarily overturning baseline-correct decisions identified on auxiliary clean training examples.
2. We develop StaCOR by combining regularized multiclass posterior correction with Selective Decision Stability, a training-time zero-margin penalty applied only to clean examples correctly predicted by the frozen noisy-support baseline. Baseline-wrong examples receive no decision-preservation constraint from this term.
3. We provide a system-level evaluation across four development benchmarks and three frozen external-validation datasets under SN and IDN protocols, covering probability quality, hard-decision stability, paired statistics, ablations, direct-context alternatives, and serving measurements. The experiments characterize when fixed-context refinement is useful and when direct clean-label context utilization remains preferable.

2. Related Work

2.1. TabPFN and Tabular Foundation Models

Prior-data fitted networks and TabPFN introduced an important route for tabular foundation modeling: rather than fitting a new predictor for each dataset, a transformer is trained across many synthetic tasks and then performs prediction by conditioning on a support set at inference time [1]. This turns the table supplied at inference into a context for amortized Bayesian-style prediction, allowing small tabular classification tasks to be solved with little or no task-specific optimization. The later Nature version substantially broadened the practical scope of this idea, reporting stronger performance on small-data tabular benchmarks and adding support for larger datasets, regression, categorical variables, missing values, reusable embeddings, density estimation, and related foundation-model capabilities [2].

Recent tabular foundation model work has extended this line in several directions. TabICL studies how tabular in-context learning can scale to larger tables through a two-stage architecture that separates row embedding from efficient in-context prediction [12]. TabSTAR explores transfer learning when tabular data include text fields, using semantically target-aware representations to connect tabular prediction with pretrained language representations [13]. EquiTabPFN addresses target-permutation equivariance, an architectural constraint that is particularly relevant when the set or order of classes changes across tasks [14]. These works indicate that tabular foundation models are moving beyond the original small-data classification setting toward larger datasets, richer metadata, and more explicit architectural invariances.

Applications are also beginning to evaluate these models in engineering domains. For example, a recent *Electronics* study compared TabPFN and TabICL for tabular intrusion detection, using cybersecurity datasets to examine how foundation models behave relative to traditional machine-learning baselines in an applied setting [15]. At the same time, robustness questions have started to receive direct attention. Nawaz et al. evaluated the robustness of the Tabular Prior-Data Fitted Network classifier in the ICML 2025 Foundation Models for Structured Data workshop, including corrupted-label conditions [16]. This is important context for the present work: existing work has begun to assess the intrinsic robustness of TabPFN under corrupted support labels. Our focus is the subsequent fixed-context refinement problem, after TabPFN has already been conditioned on a noisy support set and the deployment constraint is to leave that context unchanged. This separates robustness evaluation from output-side posterior repair.

2.2. Learning and Calibration Under Label Noise

Learning with noisy labels is a long-standing problem because corrupted supervision can bias estimation, degrade generalization, and eventually be memorized by high-capacity models. The survey of Frenay and Verleysen provides a broad taxonomy of label-noise mechanisms and learning strategies, including random, class-conditional, and instance-dependent corruption models [3]. Loss-correction approaches formalize label corruption through transition matrices and modify the training objective to compensate for class-dependent noise [17]. Sample-selection methods, such as co-teaching, use the tendency of neural networks to fit clean examples earlier than noisy ones and train paired networks that exchange small-loss examples during optimization [18].

Instance-dependent noise is especially relevant for tabular prediction because errors may depend on feature values, measurement conditions, or ambiguous cases rather than only on the true class. Xia et al. proposed part-dependent label noise as an approximation to instance-dependent corruption and modeled how different parts of an input can imply different noise transition behavior [4]. Cheng et al. studied instance-dependent label noise through a sample-sieve approach that separates likely clean and corrupted examples without requiring known noise rates [5]. These works motivate evaluating both symmetric noise and IDN in the present paper: SN gives a repeatable class-agnostic stress test, while IDN better reflects corruption processes tied to example difficulty or feature patterns.

Noisy labels also interact with probability calibration. In many pipelines, noisy supervision affects not only the final classifier but also the reliability of the posterior probabilities used for thresholding, risk scoring, or downstream decision support. Zong et al. explicitly studied prediction calibration for learning with noisy labels and proposed a Dirichlet-based calibration approach for this setting [19]. Domain-specific work in *Electronics* similarly reflects the practical importance of noisy-label learning, for example in time-series modeling for electrical power edge-end interaction, where noisy labels are handled with time-frequency representation and consistency components [20].

The present setting differs from most noisy-label learning methods because it does not retrain the base classifier, relabel the support set, or replace the noisy examples seen by TabPFN. The TabPFN backbone and the noisy support context are frozen before refinement begins. Clean evidence is used only to train and select an output-side correction head. This makes StaCOR closer to post hoc posterior refinement under label-noise-induced distortion than to conventional robust training under noisy labels.

2.3. Posterior Calibration and Decision Stability

Post hoc calibration methods modify a trained model's outputs using held-out calibration data. Temperature scaling is a simple and widely used baseline that rescales logits with a scalar temperature while preserving the argmax decision [6]. More expressive variants use ensembles, compositions, or input-dependent temperatures to improve calibration flexibility while controlling data efficiency and decision behavior [9,21]. Surveys of classifier calibration emphasize that calibration quality depends on the metric, data regime, and intended decision use, and that post hoc calibration is not a single objective independent of deployment context [22].

Dirichlet calibration is particularly relevant to this paper because it applies an affine transformation to log probabilities before a softmax, allowing class-coupled multiclass calibration [7]. Kull et al. also proposed Dirichlet calibration with off-diagonal and intercept regularization (ODIR), making it a direct precedent for parameter-level regularization of expressive multiclass calibration maps. The DC baseline in this paper should therefore be interpreted as a Dirichlet-style, ODIR-related matrix-scaling comparator rather than as an unrelated calibration family. StaCOR uses related output-side capacity-control intuition inside a frozen noisy-support TabPFN refinement setting and combines it with a separate sample-level decision-stability objective.

A parallel line of calibration work studies when a calibration map should preserve classifications or rankings. Mix-n-Match explicitly includes accuracy preservation among its calibration desiderata [9]. Intra-order-preserving calibration constrains the calibration function so that within-instance class orderings are retained [10]. Meta-Cal introduces a ranking-based framework that explicitly controls miscoverage and coverage accuracy after calibration [11]. More recent work continues this theme through accuracy-preserving probabilistic modeling on the simplex and instance-wise monotonic calibration, both of which underline that calibrated probabilities and stable decisions can be coupled by design rather than assessed only after the fact [23,24]. StaCOR uses known baseline correctness on auxiliary clean training examples and embeds a selective zero-margin boundary penalty directly into correction-head training. Unlike rank-preserving maps that constrain all samples or all class orders, this term protects only a selected subset and only against boundary violations observed during head training. That selectivity leaves room for beneficial flips on examples that the noisy-support baseline misclassifies.

Bayesian last-layer methods provide another reference point for lightweight output uncertainty modeling. Variational Bayesian Last Layers define a deterministic variational formulation for Bayesian last-layer inference that can improve uncertainty estimation and calibration in neural networks [8]. The baselines in this paper are therefore described as VBLL-inspired stochastic last-layer baselines rather than as implementations of the original VBLL training objective. This distinction matters because the formal baselines use the frozen TabPFN posterior representation, 10 Monte Carlo samples for the full MC variant, prior variance 1, initial log standard deviation -3 , and Kullback–Leibler (KL) mode none.

Related *Electronics* work also reflects the practical breadth of calibration and uncertainty estimation. Stochastic weight averaging has been studied as a confidence-calibration mechanism for deep neural networks [25]; predictive uncertainty has been used for open-

set malware traffic recognition [26]; and hybrid uncertainty calibration has been proposed for multimodal sentiment analysis under noisy or low-quality data [27]. These studies are not the same deployment setting as frozen-context TabPFN refinement, but they reinforce the broader point that calibrated probabilities, uncertainty, and decision behavior are engineering concerns in operational systems. StaCOR contributes within that landscape by jointly evaluating probability quality, harmful flips, retention, context policy, and serving cost for noisy-support TabPFN deployments.

3. Materials and Methods

StaCOR (Stability-aware Coupling-Regularized Output Refinement) is an output-side correction method for a deployment setting in which the TabPFN context is fixed and may contain noisy support labels. The method leaves the TabPFN backbone and the original context unchanged, and trains only a lightweight correction head on auxiliary clean evidence. Figure 1 provides an overview of the framework and its Selective Decision Stability mechanism.

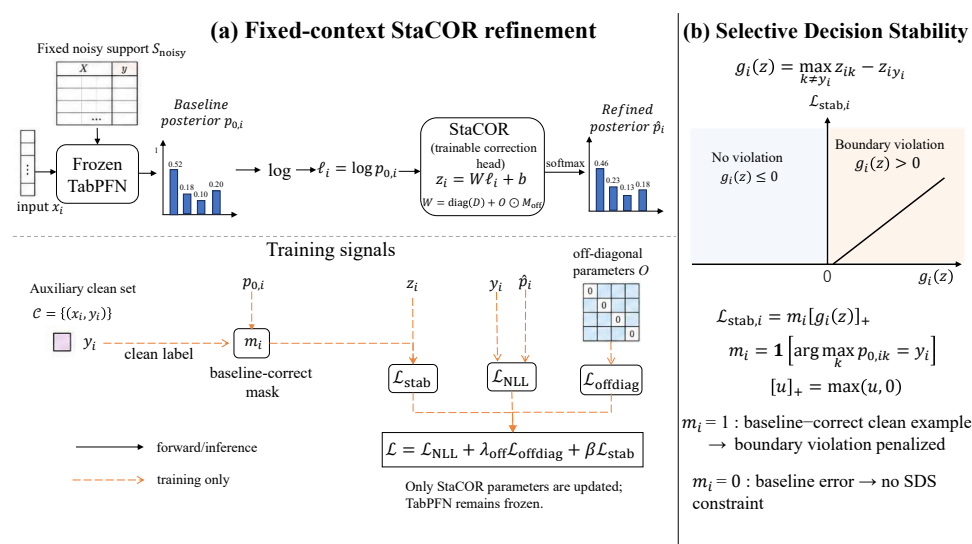


Figure 1. Overview of the StaCOR framework and Selective Decision Stability (SDS). (a) The noisy support context and TabPFN remain fixed, while the trainable StaCOR correction head refines the baseline posterior using auxiliary clean evidence for training supervision. Regularized Class Coupling provides parameter-level correction-capacity control, while the Selective Decision Stability term uses baseline correctness on clean examples to penalize decision-boundary violations. Only StaCOR parameters are updated during training; the solid forward path is retained at inference. (b) SDS applies a zero-margin penalty to baseline-correct clean examples when the refined logits cross the true-class decision boundary, whereas baseline-wrong examples receive no SDS constraint. Loss-label abbreviation: negative log-likelihood (NLL).

3.1. Problem Setting

Let S_{noisy} denote the original TabPFN support context used at serving time. Its feature vectors are observed, but its support labels may be corrupted by label noise. In addition, let $\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{n_C}$ denote a clean calibration set used only for training and selecting the output-side correction head, and let $\mathcal{Q}$ denote the held-out query or evaluation set. The query set is never used to train the correction head or to select its checkpoint.

For a query or calibration input x , the frozen TabPFN model with context S_{noisy} produces a baseline posterior

$$\mathbf{p}_0(x) = p_0(y \mid x, S_{\text{noisy}}) \in \Delta^{K-1}, \quad (1)$$

In Equation (1), K is the number of classes in the current dataset. StaCOR operates on the corresponding log-posterior vector

$$\ell(x) = \log \mathbf{p}_0(x). \quad (2)$$

Equation (2) defines the log-posterior vector. The TabPFN backbone parameters are frozen, and the noisy context S_{noisy} is assumed to remain fixed during the evaluated deployment cycle; only the output-side correction head is trainable. This is operational rather than informational: an external adaptation pipeline may use clean calibration evidence even when the served context is cached, version-locked, shared, audited, or costly to rebuild and revalidate. This distinguishes the setting from policies that directly append clean samples to the TabPFN context or replace noisy context samples with clean ones. All class-indexed vectors and matrices use the fixed canonical class order recorded in the frozen split artifact (`label_categories`). Raw dataset labels therefore need not be consecutive integers or start at zero; they are mapped into the canonical order before constructing TabPFN output vectors, correction matrices, training caches, and evaluation summaries.

3.2. Fixed-Context Output-Side Correction

StaCOR maps the baseline log posterior ℓ to corrected logits $\mathbf{z}$ through an affine class-coupled correction head. In column-vector notation, the head is

$$\mathbf{z}(x) = W\ell(x) + \mathbf{b}, \quad (3)$$

Equation (3) gives the corrected logits, followed by the corrected posterior

$$\hat{\mathbf{p}}(x) = \text{softmax}(\mathbf{z}(x)). \quad (4)$$

Using Equation (4), the final prediction is $\hat{y}(x) = \arg \max_k \hat{p}_k(x)$. Because the correction head is applied after TabPFN inference, inference still uses the original fixed noisy context; no clean calibration samples are inserted into the TabPFN support set at serving time.

For K input log-posterior components, the affine head has $K^2 + K$ parameters. This compact parameterization is appropriate for the available clean calibration regime, permits direct interpretation and regularization of cross-class entries, and adds only a lightweight serving operation. Nonlinear correction heads are possible extensions, but this study does not make a comparative claim about multilayer perceptrons.

3.3. Regularized Class Coupling

Following the general principle of off-diagonal regularization in expressive multiclass calibration [7], StaCOR uses a dedicated coupling penalty to control cross-class correction capacity in frozen noisy-support TabPFN output refinement. This parameter-level role is complementary to the sample-level decision constraint introduced in Section 3.4.

The correction matrix is decomposed into diagonal rescaling terms and off-diagonal class-coupling terms:

$$W = \text{diag}(\mathbf{D}) + O \odot M_{\text{off}}, \quad (5)$$

In Equation (5), $\mathbf{D} \in \mathbb{R}^K$, $O \in \mathbb{R}^{K \times K}$, M_{off} is zero on the diagonal and one off the diagonal, and $\odot$ denotes element-wise multiplication. The diagonal terms adjust each class log posterior independently, whereas the off-diagonal terms allow evidence for one class to affect the corrected logit of another class. With the column-vector convention in (3), W_{ij} maps the input log-posterior component for class j to the corrected output logit for class i .

The decomposition is an unconstrained reparameterization of a full affine matrix W : it neither adds expressive power nor imposes a hard structural constraint. Its utility is to

separate diagonal self-class scaling from cross-class coupling, support identity-centered initialization, and expose the off-diagonal terms for targeted regularization and interpretation.

Unrestricted full coupling can improve probability quality but may also introduce unnecessarily large cross-class perturbations. StaCOR therefore keeps the coupling capacity but regularizes its off-diagonal magnitude:

$$\mathcal{L}_{\text{offdiag}} = \frac{1}{K(K-1)} \sum_{i \neq j} W_{ij}^2. \quad (6)$$

In Equation (6), for $i \neq j$, $W_{ij} = O_{ij}$ under (5), so this is an algebraic clarification of the same loss rather than a changed objective. The term provides parameter-level control over how freely the multiclass correction map can redistribute evidence across classes while retaining the ability to model systematic class interactions.

3.4. Selective Decision Stability

Output-side calibration can improve NLL while changing hard decisions. Some changes are beneficial, but changes away from a baseline-correct prediction are harmful under the clean calibration label. StaCOR addresses this with a zero-margin reactive boundary-violation penalty defined only on calibration samples for which the frozen TabPFN baseline is already correct:

$$\mathcal{S}_{\text{correct}} = \{i : \arg \max_k \ell_{ik} = y_i\}. \quad (7)$$

Equivalently, using Equation (7), let $C^+ = \mathcal{S}_{\text{correct}}$ and define the decision-boundary violation score

$$g_i(\mathbf{z}) = \max_{k \neq y_i} z_{ik} - z_{iy_i}.$$

The condition $g_i(\mathbf{z}) \leq 0$ means that the corrected logits remain in the true-class decision region for sample i . Selective Decision Stability can be interpreted as a soft relaxation of the selective feasibility constraints

$$g_i(\mathbf{z}_\theta) \leq 0, \quad i \in C^+.$$

We implement these selective constraints through a zero-margin hinge penalty rather than an explicit constrained optimizer. For $i \in C^+$, the stability penalty is active only when the corrected true-class logit no longer exceeds the strongest wrong-class logit:

$$\mathcal{L}_{\text{stab}} = \begin{cases} \frac{1}{|C^+|} \sum_{i \in C^+} \max(0, \max_{k \neq y_i} z_{ik} - z_{iy_i}), & |C^+| > 0, \\ 0, & |C^+| = 0. \end{cases} \quad (8)$$

Thus, Equation (8) gives $\mathcal{L}_{\text{stab}} = |C^+|^{-1} \sum_{i \in C^+} [g_i(\mathbf{z})]_+$ when $C^+ \neq \emptyset$, and $\mathcal{L}_{\text{stab}} = 0$ when $C^+ = \emptyset$. This is not a positive-margin barrier: the penalty is zero for $g_i(\mathbf{z}) < 0$ and becomes positive only after the zero-margin decision constraint is violated. Samples that the baseline originally classifies incorrectly are unconstrained by SDS, so they retain the possibility of beneficial decision changes under the full soft objective. The term does not know which future query changes will be harmful; it is not a harmful-flip detector.

3.5. Training and Inference

The final StaCOR training objective is

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \lambda_{\text{off}} \mathcal{L}_{\text{offdiag}} + \beta \mathcal{L}_{\text{stab}}, \quad (9)$$

In Equation (9), $\mathcal{L}_{\text{NLL}}$ is the cross-entropy loss of the corrected logits on clean head-training samples. In the frozen final experiments, $\lambda_{\text{off}} = 1$ and $\beta = 1$. The recorded implementation setting is $\gamma = 0$: no gamma-controlled uncertainty, gate, pairwise, or allocation term is active in the final objective.

At inference time, StaCOR uses the fixed noisy context S_{noisy} , obtains the frozen TabPFN log posterior $\ell(x)$, applies the trained correction head in (3), and predicts from the corrected posterior in (4). Regularized Class Coupling and Selective Decision Stability are training-design components of the objective, not two separate inference-time modules.

3.6. Experimental Setup

3.6.1. Datasets and Evaluation Roles

The study uses four development benchmarks and three frozen external-validation datasets. The development benchmarks are Yeast [28], Wine [29], PageBlocks [30], and Optdigits [31]. Gas Sensor Drift [32], Crop Mapping [33], and Landsat Satellite [34] are used as external-validation datasets after the method design and hyperparameters were frozen. The external datasets therefore did not participate in the selection of the loss design or the fixed values $\lambda_{\text{off}} = 1$ and $\beta = 1$. Table 1 reports the dataset characteristics and evaluation roles from the frozen table artifact.

Table 1. Dataset characteristics and evaluation roles. Abbreviation: imbalance ratio (IR).

Dataset	Domain	Samples	Features	Classes	Class Imbalance	Evaluation Role
Yeast	Bio-informatics	1484	8	10	5–463 (IR = 92.60)	Development benchmark
Wine	Wine quality	4898	11	7	5–2198 (IR = 439.60)	Development benchmark
PageBlocks	Document analysis	5473	10	5	28–4913 (IR = 175.46)	Development benchmark
Optdigits	Hand-written digit recognition	5620	64	10	554–572 (IR = 1.03)	Development benchmark
Gas Sensor Drift	Chemical sensing	13,910	128	6	1641–3009 (IR = 1.83)	Frozen external validation
Crop Mapping	Remote sensing/agriculture	325,834	174	7	1143–85074 (IR = 74.43)	Frozen external validation
Landsat Satellite	Remote sensing/land cover	6435	36	6	626–1533 (IR = 2.45)	Frozen external validation

3.6.2. Noise Protocol

The benchmark protocol evaluates both symmetric noise (SN) and instance-dependent noise (IDN). For the four development benchmarks, the target noise levels are $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$, with seeds 0–4 and 20 accepted episodes per dataset–noise–noise-level–seed condition. The support and query caps are $\text{MAXSUP} = 256$ and $\text{MAXQRY} = 1024$, rather than fixed episode sizes: Yeast has 223 support and 223 query examples, while each of the other six formal datasets has 256 support and 1024 query examples. The $\eta = 0$ setting provides a clean or no-noise reference, while the main noisy-condition comparisons aggregate $\eta > 0$, i.e., $\eta \in \{0.1, 0.2, 0.3, 0.4\}$. The external-validation protocol inherits the common protocol and methods from the frozen benchmark protocol.

SN and IDN are treated as distinct perturbation mechanisms; the target η value is not interpreted as an identical realized flip rate across both noise generators.

3.6.3. Baselines

We compare StaCOR with five main baselines. Noisy Base is the frozen TabPFN model evaluated with the noisy support context and no output correction. Temperature scaling (TS) is applied to the baseline posterior [6]. DC denotes the Dirichlet calibration baseline used in the tables [7]. Its implementation takes noisy-base probabilities, clips and logs them, centers the log probabilities across classes, applies a full $K \times K$ affine map $\mathbf{z} = W\ell_{\text{centered}} + \mathbf{b}$, and returns $\text{softmax}(\mathbf{z})$. The head is initialized with $W = I$ and $\mathbf{b} = 0$, and it is trained with cross-entropy plus $10^{-4} \text{mean}((W - I)^2)$ and $10^{-4} \text{mean}(\mathbf{b}^2)$, with parameter selection by validation NLL. This is an ODIR-related Dirichlet-style baseline with full trainable coupling, diagonal shrinkage toward identity, and bias regularization; it is not the canonical off-diagonal-only ODIR objective. VBLL-D mean and VBLL-full MC are VBLL-inspired stochastic last-layer baselines [8]. VBLL-D mean uses a diagonal Gaussian stochastic last-layer parameterization and is evaluated by its posterior mean, whereas VBLL-full MC uses a fully class-coupled Gaussian parameterization with Monte Carlo predictive averaging. The formal VBLL settings use 10 Monte Carlo samples for the MC baseline, prior variance 1.0, initial log standard deviation -3.0 , and KL mode none; VBLL-D and VBLL-full differ by the diagonal-only setting.

The clean-label usage and serving experiments additionally compare direct context policies. Direct Append appends clean samples to the TabPFN context, thereby increasing the context size. Direct Replace keeps the context size fixed by replacing noisy support examples with clean ones when the policy is feasible; it is evaluated only for $B \leq 128$, because the original Yeast support has 223 examples. These direct policies are not StaCOR variants; they represent different deployment policies in which the TabPFN context can be refreshed.

3.6.4. Frozen Selective-KL Comparator

To test whether the observed trade-off is specific to SDS rather than to any auxiliary regularizer, we evaluate a frozen Selective-KL comparator. It uses the same baseline-correct subset C^+ as (7), leaves baseline-wrong examples unconstrained, and uses the detached frozen baseline posterior as the target:

$$\mathcal{L}_{\text{SKL}} = \frac{1}{|C^+|} \sum_{i \in C^+} \text{KL}(\mathbf{p}_{0,i} \| \hat{\mathbf{p}}_i), \quad |C^+| > 0. \quad (10)$$

Using the divergence in Equation (10), the comparator objective is $\mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{offdiag}} + \alpha_d \mathcal{L}_{\text{SKL}}$. For each dataset, α_d is a deterministic scale match computed as

$$\alpha_d = \frac{\mathcal{L}_{\text{SDS}}(\theta_{\text{RCC}}, \text{head_train})}{\mathcal{L}_{\text{SKL}}(\theta_{\text{RCC}}, \text{head_train})}.$$

The scale is computed from the frozen RCC-only reference checkpoint and head-training cache; it is not performance-tuned, grid-searched, or selected by evaluation metrics. The recorded $\beta = 0$ for this comparator is legacy/schema compatibility only; its active KL coefficient is `stability_weight` = α_d . Full frozen scales and results are provided in Supplementary Section S2.

3.6.5. Evaluation Metrics

Probabilistic performance is measured by negative log-likelihood (NLL), with lower values indicating better probability assignments. Decision performance is measured by

Macro-F1. To quantify whether an output-side correction disturbs useful baseline decisions, we also use harmful flip and retention. A harmful flip is a query for which the baseline prediction is correct, but the corrected prediction is wrong. A beneficial flip, used in supporting analyses, is a query for which the baseline prediction is wrong but the corrected prediction is correct. Retention is the fraction of baseline-correct predictions that remain correct after correction:

$$\text{Retention} = \frac{\#\{i : \hat{y}_{0,i} = y_i \text{ and } \hat{y}_i = y_i\}}{\#\{i : \hat{y}_{0,i} = y_i\}}, \quad (11)$$

In Equation (11), $\hat{y}_{0,i}$ is the frozen TabPFN baseline prediction and $\hat{y}_i$ is the corrected prediction. Expected calibration error (ECE) and Brier score are recorded in the frozen episode artifacts, but the main paper tables focus on NLL, Macro-F1, harmful flip, and retention.

3.6.6. Aggregation and Statistical Analysis

Main results use condition-equal and dataset-equal aggregation over noisy conditions. Episode-level metrics are averaged within each dataset–noise–noise-level condition before aggregation. For the main tables, the four noisy levels $\eta > 0$ are equally weighted within each dataset and noise kind, and datasets are then equally weighted within the development-benchmark and external-validation scopes.

Paired statistical comparisons use a 10,000-replicate paired percentile bootstrap with seed 20260815. Paired episode differences are resampled with replacement within each fixed dataset–noise–noise-level condition. Dataset identities are kept fixed rather than resampled. Condition-level estimates are subsequently aggregated with equal weighting across noisy levels and datasets. The resulting confidence intervals therefore quantify sampling variability over the fixed evaluated datasets and paired episodes, not over all possible tabular datasets.

As a read-only sensitivity analysis, we also form paired episode differences using the retry-aware key, average the 20 episodes within each dataset–kind– η –seed cluster, and resample the five seed means per condition with replacement. This sensitivity uses 10,000 percentile replicates with seed 20260815, $\eta > 0$, condition-equal and dataset-equal weighting, and fixed dataset identities; it does not replace the primary episode-level analysis. Supplementary Section S3 reports the side-by-side intervals.

3.6.7. Implementation Details

The formal protocol uses a TabPFN v2.5 classifier checkpoint recorded in the frozen protocol manifest. Each dataset is split 70/30 into calibration and final evaluation partitions. The calibration partition is then split 80/20 into head-training and head-validation partitions by proportional stratification, not class-balanced allocation. For the main calibration baselines, checkpoint or parameter selection is based on head-validation NLL. The frozen split artifact records clean calibration indices, evaluation indices, and separate head-training and head-validation indices. The head-training subset is used to fit the output-side correction heads, while the head-validation subset is used for checkpoint or parameter selection. Final query/evaluation samples are disjoint from those head-training and head-validation roles.

The primary formal evaluation environment used Python 3.10.12, PyTorch 2.5.1+cu124, TabPFN Python package 6.0.6, NumPy 1.26.4, pandas 2.3.3, and scikit-learn 1.7.2, with CUDA execution on an NVIDIA GeForce RTX 4090 graphics processing unit (GPU) (NVIDIA Corporation, Santa Clara, CA, USA). The TabPFN package version is distinct from the v2.5 classifier checkpoint specified above.

For StaCOR, the head is initialized as the identity map plus zero bias: $\mathbf{D} = \mathbf{1}$, $\mathbf{O} = 0$, and $\mathbf{b} = 0$. A separate correction head is fitted for each dataset/task; heads are not shared across datasets. Within a dataset, the same dataset-specific head is used across SN, IDN, all formal $\eta \in \{0, 0.1, 0.2, 0.3, 0.4\}$, and all formal seeds, with no per-kind, per- η , or per-seed heads. Training uses the clean head-training cache of frozen TabPFN log posteriors and clean labels, and selects the stored checkpoint by head-validation NLL on the head-validation cache. The final objective uses $\lambda_{\text{off}} = 1$, $\beta = 1$, and recorded $\gamma = 0$.

The clean-label quantity-sensitivity experiment is limited to Yeast and Optdigits. It uses budgets $B \in \{16, 32, 64, 128, 256\}$, with five clean-subset realizations at $B = 16, 32, 64, 128$ and one singleton realization at $B = 256$. The budget-specific NEW_METHOD_B heads are trained for a fixed 20 epochs without validation-based checkpoint selection. This is a clean-label quantity and clean-subset realization sensitivity study, not a distribution-shift robustness study; the complete formal protocol is in Supplementary Section S1.

The development chronology was frozen before external validation. Yeast and Optdigits were used for method discovery, which fixed ordinary off-diagonal regularization and $\lambda_{\text{off}} = 1$. Selective Decision Stability was then evaluated on head-validation evidence from Yeast and Optdigits, and $\beta = 1$ was frozen using the observed Macro-F1, harmful-flip, retention, and NLL trade-off. Wine and PageBlocks were subsequently used as confirmation datasets with $\lambda_{\text{off}} = 1$ and $\beta = 1$ already fixed, and Gas Sensor Drift, Crop Mapping, and Landsat Satellite were used only for frozen external validation. The final query/evaluation samples never participated in method development, head fitting, or hyperparameter selection.

The frozen StaCOR checkpoints record ordinary off-diagonal regularization mode, Adam optimization, learning rate 0.001, 20 training epochs, batch size 256, train seed 0, and data-order seed 0. No weight-decay field is recorded, and the training implementation instantiates Adam without a weight-decay argument.

4. Results

4.1. Probabilistic Performance

Table 2 summarizes NLL across the seven final datasets. On the development benchmarks under SN, StaCOR and DC are nearly identical in average NLL: 0.7594 for StaCOR versus 0.7592 for DC. Under IDN, DC has the lower benchmark average NLL, 0.6720 versus 0.6738 for StaCOR. Thus, the NLL comparison with DC should be interpreted as close under benchmark SN and slightly unfavorable to StaCOR under benchmark IDN.

Relative to the VBLL baselines, StaCOR is competitive or better in the aggregate NLL columns. The paired bootstrap comparison supports lower benchmark NLL for StaCOR than for VBLL-full MC under both SN (difference -0.00295 , 95% confidence interval (CI) $[-0.00367, -0.00223]$) and IDN (difference -0.00425 , 95% CI $[-0.00504, -0.00348]$). On the external-validation datasets, DC again has slightly lower average NLL than StaCOR under both SN and IDN, while StaCOR improves over VBLL-D mean and VBLL-full MC in the external averages. These results support StaCOR as an output-side correction with competitive probabilistic performance, but its main motivation is the probability–decision trade-off examined next.

Table 2. Probabilistic performance across seven datasets. Lower negative log-likelihood (NLL) is better. Boldface marks the lowest numerical value within each panel and dataset or aggregate column; it is metric-specific rather than an overall method ranking. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); temperature scaling (TS); Dirichlet calibration (DC); Variational Bayesian Last Layers (VBLL); Monte Carlo (MC). Development benchmarks.

Panel	Method	Yeast	Wine	PageBlocks	Optdigits	Benchmark Avg.
Panel A: SN	Noisy Base	1.3340	1.2340	0.4185	0.6885	0.9187
Panel A: SN	TS	1.2772	1.2052	0.2540	0.4397	0.7940
Panel A: SN	DC	1.2297	1.1304	0.2419	0.4347	0.7592
Panel A: SN	VBLL-D mean	1.2401	1.1508	0.2420	0.4492	0.7705
Panel A: SN	VBLL-full MC	1.2262	1.1301	0.2443	0.4488	0.7624
Panel A: SN	StaCOR	1.2223	1.1402	0.2420	0.4331	0.7594
Panel B: IDN	Noisy Base	1.1715	1.1896	0.2961	0.3233	0.7451
Panel B: IDN	TS	1.1542	1.1816	0.1958	0.2640	0.6989
Panel B: IDN	DC	1.1291	1.1153	0.1935	0.2503	0.6720
Panel B: IDN	VBLL-D mean	1.1385	1.1317	0.1950	0.2483	0.6784
Panel B: IDN	VBLL-full MC	1.1476	1.1162	0.1946	0.2537	0.6780
Panel B: IDN	StaCOR	1.1307	1.1257	0.1930	0.2459	0.6738

Panel	Method	Gas	Crop	Landsat	External Avg.
Panel A: SN	Noisy Base	0.5707	0.4858	0.7183	0.5916
Panel A: SN	TS	0.3575	0.2396	0.5791	0.3921
Panel A: SN	DC	0.3564	0.2211	0.5413	0.3729
Panel A: SN	VBLL-D mean	0.3656	0.2352	0.5488	0.3832
Panel A: SN	VBLL-full MC	0.3633	0.2354	0.5396	0.3794
Panel A: SN	StaCOR	0.3587	0.2263	0.5373	0.3741
Panel B: IDN	Noisy Base	0.3154	0.2612	0.5241	0.3669
Panel B: IDN	TS	0.2141	0.1474	0.4602	0.2739
Panel B: IDN	DC	0.2089	0.1410	0.4413	0.2637
Panel B: IDN	VBLL-D mean	0.2100	0.1437	0.4456	0.2665
Panel B: IDN	VBLL-full MC	0.2103	0.1458	0.4409	0.2657
Panel B: IDN	StaCOR	0.2096	0.1429	0.4402	0.2643

4.2. Decision Performance and Probability–Decision Trade-Off

Table 3 reports Macro-F1, harmful flip, and retention. On the development benchmarks under SN, StaCOR has a benchmark-average harmful-flip rate of 0.0124, lower than DC at 0.0195 and VBLL-full MC at 0.0331. Its retention is 0.9781, higher than DC at 0.9655. Under IDN, the same pattern holds against DC: StaCOR has a harmful flip of 0.0124 versus 0.0202, and retention of 0.9781 versus 0.9628.

External validation shows the same decision-stability emphasis, while also showing the Macro-F1 trade-off against DC. Under SN, DC has the higher external-average Macro-F1, 0.7935 versus 0.7907 for StaCOR, but StaCOR has lower harmful flip, 0.0042 versus 0.0090, and higher retention, 0.9951 versus 0.9894. Under IDN, DC again has the higher external-average Macro-F1, 0.8440 versus 0.8426, while StaCOR has lower harmful flip, 0.0027 versus 0.0062, and higher retention, 0.9969 versus 0.9930.

Figure 2 plots NLL against harmful flip for the four benchmark/external and SN/IDN panels. The plotted front is an empirical Pareto front among the compared methods, not a theoretical frontier. The figure illustrates the central trade-off: probability-oriented calibration can lower NLL while increasing harmful decision changes, whereas StaCOR is designed to obtain a more favorable balance between probability quality and decision stability.

Table 3. Decision performance across seven datasets. Macro-F1 and retention are higher-is-better; harmful flip is lower-is-better. Boldface marks the numerical optimum within each panel, metric, and dataset or aggregate column; it is metric-specific rather than an overall method ranking. For Noisy Base and temperature scaling (TS), zero harmful flip and full retention are structural consequences of retaining the noisy-base decision and do not indicate overall predictive superiority. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); Dirichlet calibration (DC); Variational Bayesian Last Layers (VBLL); Monte Carlo (MC). Development benchmarks.

Panel	Metric	Method	Yeast	Wine	PageBlocks	Optdigits	Benchmark Avg.
Panel A: SN	Macro-F1	Noisy Base	0.3104	0.1900	0.4008	0.8673	0.4421
Panel A: SN	Macro-F1	TS	0.3104	0.1900	0.4008	0.8673	0.4421
Panel A: SN	Macro-F1	DC	0.3047	0.1804	0.3920	0.8667	0.4359
Panel A: SN	Macro-F1	VBLL-D mean	0.2845	0.1684	0.3830	0.8671	0.4257
Panel A: SN	Macro-F1	VBLL-full MC	0.2802	0.1744	0.3871	0.8647	0.4266
Panel A: SN	Macro-F1	StaCOR	0.2994	0.1854	0.3947	0.8688	0.4371
Panel A: SN	Harmful Flip	Noisy Base	0.0000	0.0000	0.0000	0.0000	0.0000
Panel A: SN	Harmful Flip	TS	0.0000	0.0000	0.0000	0.0000	0.0000
Panel A: SN	Harmful Flip	DC	0.0168	0.0450	0.0011	0.0150	0.0195
Panel A: SN	Harmful Flip	VBLL-D mean	0.0162	0.0423	0.0018	0.0064	0.0167
Panel A: SN	Harmful Flip	VBLL-full MC	0.0573	0.0559	0.0024	0.0168	0.0331
Panel A: SN	Harmful Flip	StaCOR	0.0232	0.0177	0.0014	0.0072	0.0124
Panel A: SN	Retention	Noisy Base	1.0000	1.0000	1.0000	1.0000	1.0000
Panel A: SN	Retention	TS	1.0000	1.0000	1.0000	1.0000	1.0000
Panel A: SN	Retention	DC	0.9694	0.9115	0.9989	0.9822	0.9655
Panel A: SN	Retention	VBLL-D mean	0.9700	0.9163	0.9981	0.9923	0.9692
Panel A: SN	Retention	VBLL-full MC	0.8944	0.8903	0.9974	0.9800	0.9406
Panel A: SN	Retention	StaCOR	0.9572	0.9651	0.9985	0.9914	0.9781
Panel B: IDN	Macro-F1	Noisy Base	0.3746	0.2136	0.4717	0.9257	0.4964
Panel B: IDN	Macro-F1	TS	0.3746	0.2136	0.4717	0.9257	0.4964
Panel B: IDN	Macro-F1	DC	0.3697	0.2079	0.4565	0.9257	0.4899
Panel B: IDN	Macro-F1	VBLL-D mean	0.3491	0.1935	0.4487	0.9257	0.4792
Panel B: IDN	Macro-F1	VBLL-full MC	0.3294	0.2023	0.4477	0.9251	0.4761
Panel B: IDN	Macro-F1	StaCOR	0.3661	0.2082	0.4625	0.9266	0.4908
Panel B: IDN	Harmful Flip	Noisy Base	0.0000	0.0000	0.0000	0.0000	0.0000
Panel B: IDN	Harmful Flip	TS	0.0000	0.0000	0.0000	0.0000	0.0000
Panel B: IDN	Harmful Flip	DC	0.0221	0.0483	0.0014	0.0089	0.0202
Panel B: IDN	Harmful Flip	VBLL-D mean	0.0143	0.0301	0.0021	0.0023	0.0122
Panel B: IDN	Harmful Flip	VBLL-full MC	0.0631	0.0590	0.0029	0.0095	0.0336
Panel B: IDN	Harmful Flip	StaCOR	0.0259	0.0176	0.0016	0.0045	0.0124
Panel B: IDN	Retention	Noisy Base	1.0000	1.0000	1.0000	1.0000	1.0000
Panel B: IDN	Retention	TS	1.0000	1.0000	1.0000	1.0000	1.0000
Panel B: IDN	Retention	DC	0.9617	0.9006	0.9985	0.9903	0.9628
Panel B: IDN	Retention	VBLL-D mean	0.9751	0.9380	0.9978	0.9975	0.9771
Panel B: IDN	Retention	VBLL-full MC	0.8908	0.8787	0.9970	0.9898	0.9391
Panel B: IDN	Retention	StaCOR	0.9552	0.9636	0.9983	0.9951	0.9781
Panel	Metric	Method	Gas	Crop	Landsat	External Avg.	
Panel A: SN	Macro-F1	Noisy Base	0.8816	0.7413	0.7540	0.7923	
Panel A: SN	Macro-F1	TS	0.8816	0.7413	0.7540	0.7923	
Panel A: SN	Macro-F1	DC	0.8806	0.7378	0.7623	0.7935	
Panel A: SN	Macro-F1	VBLL-D mean	0.8805	0.7320	0.7530	0.7885	
Panel A: SN	Macro-F1	VBLL-full MC	0.8797	0.7233	0.7507	0.7846	
Panel A: SN	Macro-F1	StaCOR	0.8817	0.7356	0.7549	0.7907	
Panel A: SN	Harmful Flip	Noisy Base	0.0000	0.0000	0.0000	0.0000	
Panel A: SN	Harmful Flip	TS	0.0000	0.0000	0.0000	0.0000	
Panel A: SN	Harmful Flip	DC	0.0097	0.0063	0.0109	0.0090	
Panel A: SN	Harmful Flip	VBLL-D mean	0.0039	0.0035	0.0065	0.0046	
Panel A: SN	Harmful Flip	VBLL-full MC	0.0111	0.0069	0.0116	0.0099	
Panel A: SN	Harmful Flip	StaCOR	0.0040	0.0031	0.0054	0.0042	
Panel A: SN	Retention	Noisy Base	1.0000	1.0000	1.0000	1.0000	
Panel A: SN	Retention	TS	1.0000	1.0000	1.0000	1.0000	
Panel A: SN	Retention	DC	0.9886	0.9931	0.9865	0.9894	
Panel A: SN	Retention	VBLL-D mean	0.9954	0.9962	0.9919	0.9945	
Panel A: SN	Retention	VBLL-full MC	0.9870	0.9925	0.9856	0.9884	
Panel A: SN	Retention	StaCOR	0.9953	0.9966	0.9934	0.9951	
Panel B: IDN	Macro-F1	Noisy Base	0.9355	0.8039	0.7933	0.8442	
Panel B: IDN	Macro-F1	TS	0.9355	0.8039	0.7933	0.8442	
Panel B: IDN	Macro-F1	DC	0.9353	0.7967	0.8000	0.8440	
Panel B: IDN	Macro-F1	VBLL-D mean	0.9353	0.7977	0.7871	0.8400	
Panel B: IDN	Macro-F1	VBLL-full MC	0.9348	0.7750	0.7891	0.8330	
Panel B: IDN	Macro-F1	StaCOR	0.9353	0.8005	0.7920	0.8426	
Panel B: IDN	Harmful Flip	Noisy Base	0.0000	0.0000	0.0000	0.0000	

Table 3. Cont.

Panel	Metric	Method	Gas	Crop	Landsat	External Avg.
Panel B: IDN	Harmful Flip	TS	0.0000	0.0000	0.0000	0.0000
Panel B: IDN	Harmful Flip	DC	0.0047	0.0036	0.0103	0.0062
Panel B: IDN	Harmful Flip	VBLL-D mean	0.0015	0.0014	0.0064	0.0031
Panel B: IDN	Harmful Flip	VBLL-full MC	0.0054	0.0044	0.0099	0.0066
Panel B: IDN	Harmful Flip	StaCOR	0.0019	0.0017	0.0045	0.0027
Panel B: IDN	Retention	Noisy Base	1.0000	1.0000	1.0000	1.0000
Panel B: IDN	Retention	TS	1.0000	1.0000	1.0000	1.0000
Panel B: IDN	Retention	DC	0.9949	0.9962	0.9878	0.9930
Panel B: IDN	Retention	VBLL-D mean	0.9984	0.9985	0.9924	0.9964
Panel B: IDN	Retention	VBLL-full MC	0.9942	0.9954	0.9882	0.9926
Panel B: IDN	Retention	StaCOR	0.9980	0.9982	0.9946	0.9969

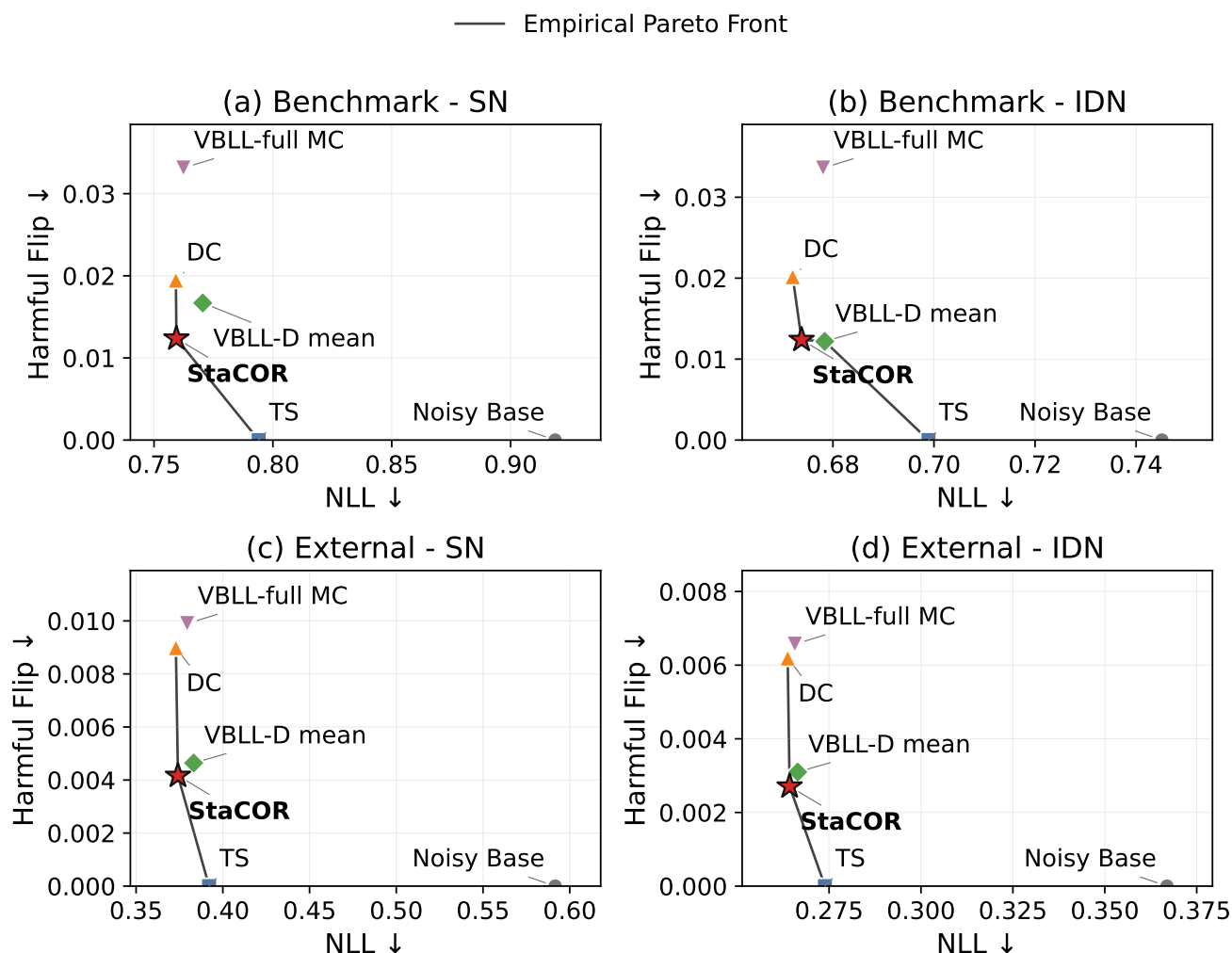


Figure 2. Probability-decision Pareto comparison. Each panel plots aggregated negative log-likelihood (NLL) against harmful flip for the compared methods; lower values on both axes are preferred. The front shown is empirical within the compared method set. Downward arrows indicate that lower values are better. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); temperature scaling (TS); Dirichlet calibration (DC); Variational Bayesian Last Layers (VBLL); Monte Carlo (MC).

4.3. Selective Decision Stability Ablation

Table 4 and Figure 3 compare the final $\beta = 1$ StaCOR objective against the capacity-only $\beta = 0$ variant. The deltas are computed as final StaCOR minus capacity-only. Positive Macro-F1 deltas and retention deltas therefore favor $\beta = 1$, while negative harmful-flip deltas favor $\beta = 1$.

Table 4. Selective Decision Stability ablation. Deltas are final StaCOR minus the capacity-only $\beta = 0$ variant, with 95% paired percentile bootstrap intervals. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); negative log-likelihood (NLL). Probability and classification metrics.

Dataset	Noise	Delta NLL	Delta Macro-F1
Yeast	SN	−0.00023 [−0.00058, +0.00013]	+0.00438 [+0.00295, +0.00588]
Yeast	IDN	−0.00268 [−0.00326, −0.00213]	+0.00760 [+0.00522, +0.01002]
Wine	SN	+0.00138 [+0.00105, +0.00171]	+0.00575 [+0.00476, +0.00677]
Wine	IDN	+0.00445 [+0.00387, +0.00505]	+0.00474 [+0.00373, +0.00577]
PageBlocks	SN	+0.00007 [−0.00004, +0.00017]	+0.00334 [+0.00245, +0.00432]
PageBlocks	IDN	−0.00072 [−0.00081, −0.00062]	+0.00619 [+0.00480, +0.00767]
Optdigits	SN	+0.00009 [−0.00009, +0.00027]	+0.00028 [+0.00010, +0.00046]
Optdigits	IDN	+0.00006 [−0.00007, +0.00019]	+0.00018 [+0.00004, +0.00031]
Gas	SN	−0.00026 [−0.00059, +0.00006]	+0.00104 [+0.00066, +0.00144]
Gas	IDN	−0.00011 [−0.00032, +0.00012]	+0.00023 [+0.00003, +0.00044]
Crop	SN	+0.00016 [+0.00006, +0.00026]	+0.00214 [+0.00137, +0.00302]
Crop	IDN	−0.00012 [−0.00019, −0.00004]	+0.00196 [+0.00103, +0.00302]
Landsat	SN	+0.00057 [+0.00041, +0.00074]	+0.00452 [+0.00401, +0.00502]
Landsat	IDN	−0.00040 [−0.00054, −0.00025]	+0.00369 [+0.00327, +0.00413]
Dataset	Noise	Delta Harmful	Delta Retention
Yeast	SN	−0.00659 [−0.00738, −0.00583]	+0.01201 [+0.01058, +0.01348]
Yeast	IDN	−0.00900 [−0.00996, −0.00807]	+0.01556 [+0.01396, +0.01719]
Wine	SN	−0.02313 [−0.02478, −0.02148]	+0.04549 [+0.04214, +0.04889]
Wine	IDN	−0.02440 [−0.02562, −0.02325]	+0.04990 [+0.04734, +0.05252]
PageBlocks	SN	−0.00032 [−0.00039, −0.00025]	+0.00034 [+0.00027, +0.00041]
PageBlocks	IDN	−0.00045 [−0.00053, −0.00037]	+0.00048 [+0.00040, +0.00056]
Optdigits	SN	−0.00134 [−0.00147, −0.00121]	+0.00158 [+0.00143, +0.00174]
Optdigits	IDN	−0.00090 [−0.00100, −0.00081]	+0.00097 [+0.00087, +0.00108]
Gas	SN	−0.00148 [−0.00171, −0.00127]	+0.00174 [+0.00149, +0.00200]
Gas	IDN	−0.00058 [−0.00071, −0.00045]	+0.00062 [+0.00048, +0.00075]
Crop	SN	−0.00057 [−0.00067, −0.00048]	+0.00062 [+0.00053, +0.00072]
Crop	IDN	−0.00030 [−0.00037, −0.00024]	+0.00032 [+0.00025, +0.00039]
Landsat	SN	−0.00179 [−0.00200, −0.00157]	+0.00216 [+0.00190, +0.00244]
Landsat	IDN	−0.00203 [−0.00222, −0.00186]	+0.00242 [+0.00220, +0.00264]

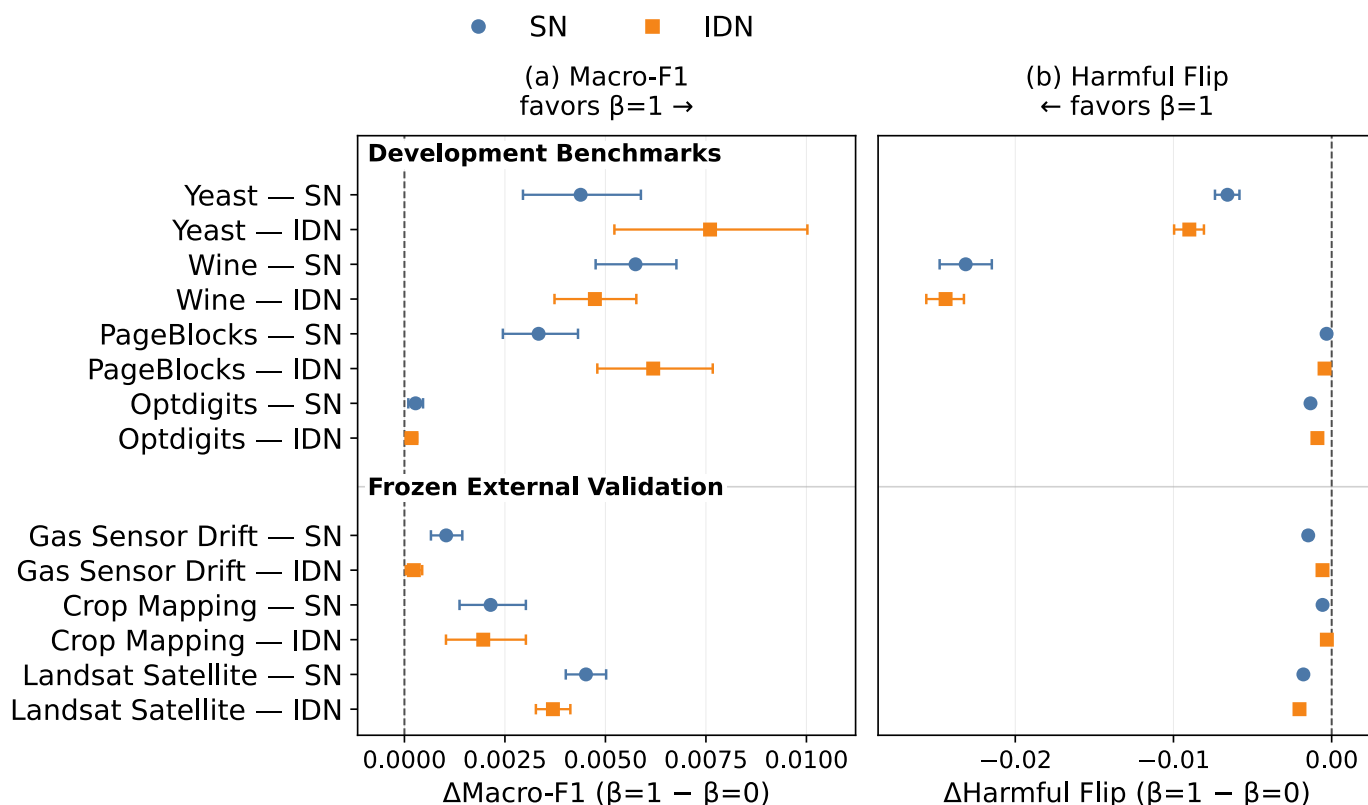


Figure 3. Selective Decision Stability ablation. Positive Macro-F1 deltas and negative harmful-flip deltas favor the final $\beta = 1$ objective. The arrows indicate these favorable directions; the dashed vertical lines mark zero difference. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); confidence interval (CI).

On the four development benchmarks, Macro-F1 improves in all eight dataset–noise cells, harmful flip decreases in all eight cells, and retention increases in all eight cells. The frozen external-validation datasets show the same directional consistency across all six dataset–noise cells. These results support a contribution of Selective Decision Stability to the observed decision-protection behavior.

The ablation also indicates small NLL costs in some settings. NLL deltas are positive in five of eight development cells and two of six external-validation cells, including Wine SN and IDN, Crop SN, and Landsat SN, with confidence intervals above zero. Other cells show neutral or negative NLL deltas. The appropriate interpretation is therefore that Selective Decision Stability consistently improves the decision metrics measured here, with small measurable NLL costs in some settings.

4.4. Paired Statistical Comparison

Table 5 reports paired differences against DC, VBLL-D mean, and VBLL-full MC. On the development benchmarks under SN, the StaCOR–DC NLL difference is +0.00022 with 95% CI $[-0.00025, +0.00068]$, so the paired interval does not reliably separate the two methods in NLL. In the same scope, StaCOR improves Macro-F1 by +0.00112, reduces harmful flip by -0.00710 , and improves retention by +0.01259.

Table 5. Paired statistical comparison against key baselines. Differences are StaCOR minus the reference method. For negative log-likelihood (NLL) and harmful flip, negative values favor StaCOR; for Macro-F1 and retention, positive values favor StaCOR. Abbreviations: symmetric noise (SN); instance-dependent noise (IDN); Dirichlet calibration (DC); Variational Bayesian Last Layers (VBLL); Monte Carlo (MC); confidence interval (CI). Probability and classification metrics.

Comparison	Delta NLL	Delta Macro–F1
Benchmark—SN		
StaCOR–DC	+0.00022 [−0.00025, +0.00068]	+0.00112 [+0.00040, +0.00184]
StaCOR–VBLL – D mean	−0.01108 [−0.01146, −0.01069]	+0.01133 [+0.01047, +0.01219]
StaCOR–VBLL – full MC	−0.00295 [−0.00367, −0.00223]	+0.01047 [+0.00932, +0.01165]
Benchmark—IDN		
StaCOR–DC	+0.00176 [+0.00124, +0.00228]	+0.00091 [+0.00005, +0.00177]
StaCOR–VBLL – D mean	−0.00458 [−0.00507, −0.00407]	+0.01162 [+0.01062, +0.01266]
StaCOR–VBLL – full MC	−0.00425 [−0.00504, −0.00348]	+0.01473 [+0.01335, +0.01615]
External Validation—SN		
StaCOR–DC	+0.00119 [+0.00053, +0.00182]	−0.00282 [−0.00383, −0.00177]
StaCOR–VBLL – D mean	−0.00904 [−0.00943, −0.00866]	+0.00222 [+0.00172, +0.00274]
StaCOR–VBLL – full MC	−0.00532 [−0.00597, −0.00463]	+0.00616 [+0.00490, +0.00752]
External Validation—IDN		
StaCOR–DC	+0.00053 [+0.00000, +0.00104]	−0.00143 [−0.00239, −0.00046]
StaCOR–VBLL – D mean	−0.00220 [−0.00249, −0.00191]	+0.00256 [+0.00210, +0.00306]
StaCOR–VBLL – full MC	−0.00142 [−0.00196, −0.00088]	+0.00960 [+0.00792, +0.01133]
Comparison	Delta Harmful	Delta Retention
Benchmark—SN		
StaCOR–DC	−0.00710 [−0.00766, −0.00656]	+0.01259 [+0.01153, +0.01369]
StaCOR–VBLL – D mean	−0.00432 [−0.00486, −0.00378]	+0.00887 [+0.00780, +0.00992]
StaCOR–VBLL – full MC	−0.02075 [−0.02182, −0.01971]	+0.03752 [+0.03560, +0.03957]
Benchmark—IDN		
StaCOR–DC	−0.00779 [−0.00825, −0.00734]	+0.01526 [+0.01435, +0.01617]
StaCOR–VBLL – D mean	+0.00020 [−0.00021, +0.00062]	+0.00096 [+0.00020, +0.00170]
StaCOR–VBLL – full MC	−0.02122 [−0.02223, −0.02025]	+0.03898 [+0.03721, +0.04081]
External Validation—SN		
StaCOR–DC	−0.00483 [−0.00510, −0.00456]	+0.00567 [+0.00535, +0.00598]
StaCOR–VBLL – D mean	−0.00048 [−0.00063, −0.00033]	+0.00058 [+0.00040, +0.00077]
StaCOR–VBLL – full MC	−0.00574 [−0.00607, −0.00541]	+0.00672 [+0.00635, +0.00711]
External Validation—IDN		
StaCOR–DC	−0.00349 [−0.00367, −0.00330]	+0.00394 [+0.00373, +0.00415]
StaCOR–VBLL – D mean	−0.00039 [−0.00051, −0.00028]	+0.00049 [+0.00036, +0.00063]
StaCOR–VBLL – full MC	−0.00386 [−0.00406, −0.00366]	+0.00432 [+0.00410, +0.00454]

Under benchmark IDN, StaCOR has a small but statistically supported increase in NLL relative to DC: $+0.00176$ with CI $[+0.00124, +0.00228]$. The decision metrics still favor StaCOR in this paired comparison, with lower harmful flip and higher retention. On the external-validation datasets, DC also has lower NLL and higher Macro-F1 in the paired comparisons under both SN and IDN, while StaCOR retains consistent harmful-flip and retention advantages. This pattern reinforces that StaCOR is not a universal NLL winner over DC; its advantage is concentrated in decision stability.

Against VBLL-full MC, StaCOR has lower NLL, higher Macro-F1, lower harmful flip, and higher retention in both benchmark noise regimes and both external-validation noise regimes. Against VBLL-D mean, the comparisons are also mostly favorable for StaCOR, with the main exception that the benchmark IDN harmful-flip interval crosses zero.

4.5. Frozen Selective-KL Comparator and Seed-Clustered Sensitivity

The frozen Selective-KL comparator isolates an alternative selective regularizer under the same fixed-context protocol. Across all four development/external and SN/IDN scopes, SDS has higher Macro-F1, lower harmful flip, and higher retention than scale-matched Selective-KL. For NLL, SDS is significantly lower under development SN and external SN; the development-IDN and external-IDN paired intervals include zero. Thus, under the evaluated fixed-context protocol, SDS provides a more favorable probability–decision trade-off than this scale-matched Selective-KL comparator, rather than establishing superiority over alternative objectives generally.

The primary paired episode-level bootstrap remains unchanged. In the independent seed-clustered sensitivity analysis, all 64 point-estimate directions and all 64 zero-exclusion conclusions agree with the episode-level reference; no previously zero-excluding interval becomes zero-including, and 15 of 64 seed-clustered intervals are wider. These results support robustness of the reported conclusions to this more conservative five-seed-cluster sensitivity analysis, without asserting independence of episodes or clusters. Full comparator and side-by-side interval results are in Supplementary Sections S2 and S3.

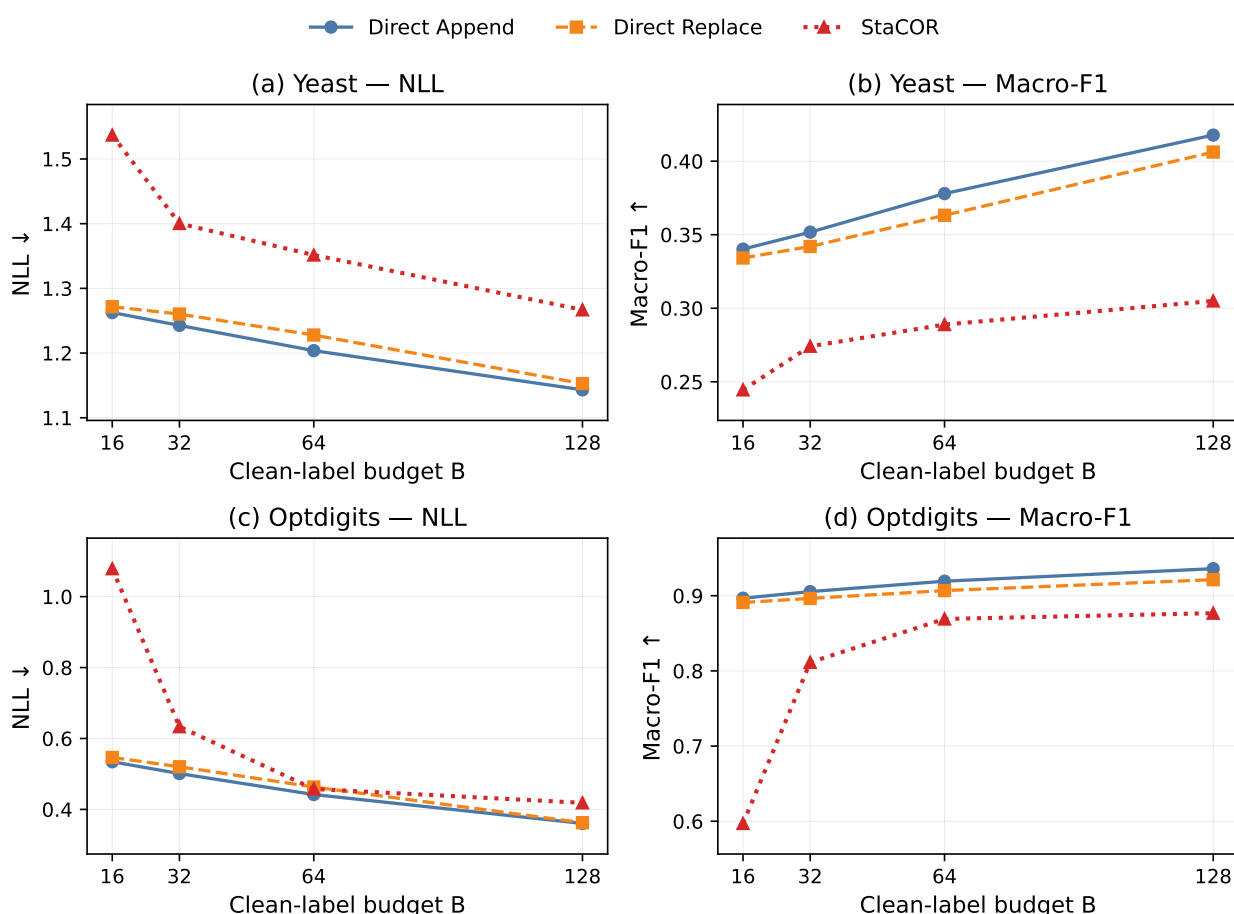
4.6. Clean-Label Usage and Context Policies

Table 6 and Figure 4 evaluate how the same clean-label subset and budget B are used under different deployment policies on Yeast and Optsdigits. Direct Append appends the clean subset C_B to the TabPFN context, Direct Replace uses the same C_B to replace B noisy support samples when feasible so that context size remains fixed, and StaCOR keeps the original noisy context unchanged while using the same C_B to train the output-side correction head. When clean samples can be directly incorporated into the TabPFN context, direct context utilization generally outperforms output-side correction.

For example, on Yeast with budget 128, Direct Append reaches NLL 1.1431 and Macro-F1 0.4177, while StaCOR has NLL 1.2668 and Macro-F1 0.3050. On Optsdigits with budget 128, Direct Append reaches NLL 0.3604 and Macro-F1 0.9360, compared with StaCOR at NLL 0.4186 and Macro-F1 0.8768. There are local metric-specific exceptions, such as StaCOR having fewer harmful flips than Direct Replace in some budget cells, but the overall conclusion is that StaCOR is most appropriate when the original TabPFN context is fixed, cached, version-locked, or expensive to refresh.

Table 6. Clean-label usage under equal clean-label budgets for different context policies. Abbreviation: negative log-likelihood (NLL).

Dataset	Budget	Method	NLL	Macro-F1	Harmful	Retention
Yeast	16	Direct Append	1.2626	0.3402	0.0221	0.9597
Yeast	16	Direct Replace	1.2716	0.3342	0.0356	0.9350
Yeast	16	StaCOR	1.5369	0.2445	0.1136	0.7954
Yeast	64	Direct Append	1.2038	0.3779	0.0351	0.9357
Yeast	64	Direct Replace	1.2279	0.3632	0.0630	0.8852
Yeast	64	StaCOR	1.3515	0.2889	0.0639	0.8848
Yeast	128	Direct Append	1.1431	0.4177	0.0451	0.9174
Yeast	128	Direct Replace	1.1528	0.4062	0.0843	0.8466
Yeast	128	StaCOR	1.2668	0.3050	0.0441	0.9200
Optdigits	16	Direct Append	0.5344	0.8967	0.0081	0.9905
Optdigits	16	Direct Replace	0.5465	0.8910	0.0174	0.9796
Optdigits	16	StaCOR	1.0788	0.5971	0.2369	0.7310
Optdigits	64	Direct Append	0.4419	0.9193	0.0092	0.9893
Optdigits	64	Direct Replace	0.4632	0.9068	0.0260	0.9700
Optdigits	64	StaCOR	0.4577	0.8691	0.0280	0.9672
Optdigits	128	Direct Append	0.3604	0.9360	0.0088	0.9899
Optdigits	128	Direct Replace	0.3625	0.9214	0.0290	0.9669
Optdigits	128	StaCOR	0.4186	0.8768	0.0183	0.9785

**Figure 4.** Clean-label budget and context-policy comparison on Yeast and Optdigits. Direct Append and Direct Replace use clean samples through the TabPFN context, whereas StaCOR keeps the original context fixed and applies output-side correction. Downward arrows for negative log-likelihood (NLL) indicate that lower values are better; upward arrows for Macro-F1 indicate that higher values are better.

4.7. Serving Efficiency

Table 7 and Figure 5 compare the serving context and measured inference cost for Noisy Base, Direct Append, Direct Replace, and StaCOR. StaCOR keeps the original serving context size fixed: 223 for Yeast and 256 for Optdigits in Table 7. Its correction-head latency is small relative to total measured inference latency. For example, Optdigits at budget 128 reports adapter latency 0.238 ms and steady-state latency 240.67 ms, approximately 0.1% of the measured steady-state latency.

Direct Append changes the serving context as the clean-label budget grows. In Table 7, Optdigits budget 128 increases the context size from 256 to 384 and incremental peak GPU memory from 181.31 MB for StaCOR to 191.91 MB for Direct Append. Its steady-state latency is 265.71 ms, about 10.4% higher than the corresponding StaCOR value in the same table, and its refresh-plus-first-predict latency is 408.01 ms versus 352.49 ms. Figure 5 includes the broader budget curve, where the latency effect becomes larger for Optdigits at higher appended budgets.

The Yeast measurements show why the latency conclusion should remain workload-dependent. Direct Append changes the context size and memory footprint, but its measured steady-state latency differs only slightly from StaCOR in the displayed Yeast budget cells. Thus, context expansion consistently changes serving context and memory requirements, while latency impact depends on the workload and budget.

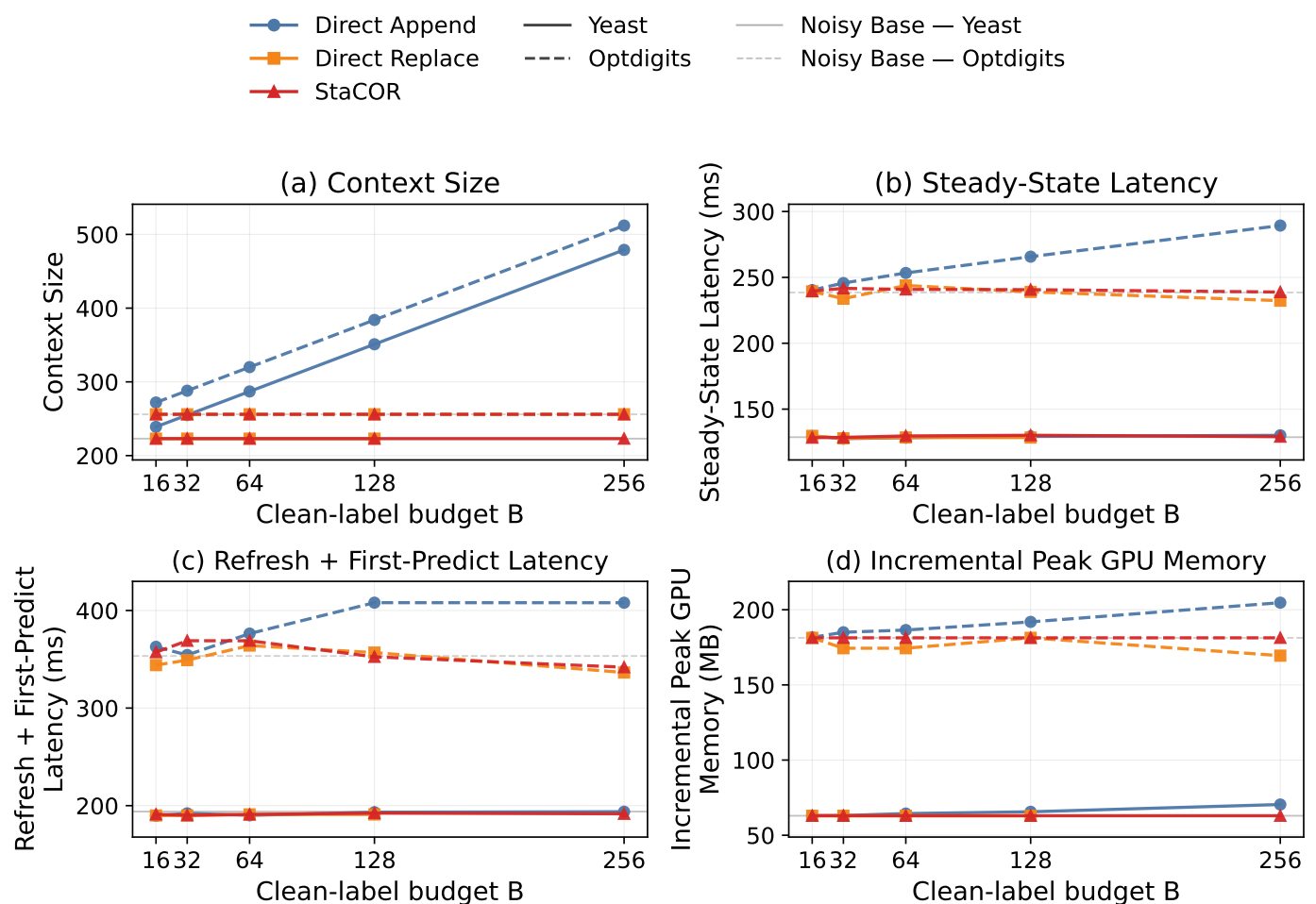


Figure 5. Serving efficiency and context scaling. Direct Append increases context size with clean-label budget, whereas StaCOR keeps the original serving context fixed and adds only the correction-head latency. Abbreviation: graphics processing unit (GPU). MB denotes megabytes.

Table 7. Serving context and inference cost. Abbreviation: graphics processing unit (GPU). MB denotes megabytes. Context, steady-state and adapter latency.

Dataset	Budget	Method	Context Size	Steady-State Latency (ms)	Adapter Latency (ms)
Yeast	N/A	Noisy Base	223	128.72	0.000
Yeast	16	Direct Append	239	129.57	0.000
Yeast	16	Direct Replace	223	129.70	0.000
Yeast	16	StaCOR	223	128.34	0.148
Yeast	128	Direct Append	351	129.33	0.000
Yeast	128	Direct Replace	223	128.43	0.000
Yeast	128	StaCOR	223	130.16	0.152
Optdigits	N/A	Noisy Base	256	238.55	0.000
Optdigits	16	Direct Append	272	240.35	0.000
Optdigits	16	Direct Replace	256	239.34	0.000
Optdigits	16	StaCOR	256	239.54	0.244
Optdigits	128	Direct Append	384	265.71	0.000
Optdigits	128	Direct Replace	256	238.99	0.000
Optdigits	128	StaCOR	256	240.67	0.238

Dataset	Budget	Method	Refresh + First-Predict Latency (ms)	Incremental Peak GPU Memory (MB)
Yeast	N/A	Noisy Base	193.92	62.96
Yeast	16	Direct Append	189.64	63.02
Yeast	16	Direct Replace	189.92	62.96
Yeast	16	StaCOR	191.15	62.96
Yeast	128	Direct Append	193.26	65.60
Yeast	128	Direct Replace	190.97	62.96
Yeast	128	StaCOR	192.33	62.96
Optdigits	N/A	Noisy Base	353.38	181.31
Optdigits	16	Direct Append	362.78	181.62
Optdigits	16	Direct Replace	344.04	181.31
Optdigits	16	StaCOR	357.22	181.31
Optdigits	128	Direct Append	408.01	191.91
Optdigits	128	Direct Replace	356.85	181.30
Optdigits	128	StaCOR	352.49	181.31

5. Discussion

5.1. Probability Quality Versus Decision Stability

The results indicate that probabilistic quality and decision stability should be treated as related but distinct evaluation targets. NLL evaluates the probability assigned to the true label over the full predictive distribution, whereas Macro-F1 depends only on the final hard decision after taking the argmax. Harmful flip and retention focus on a still narrower question: whether a correction changes predictions that the noisy-support TabPFN baseline had already classified correctly. A post hoc correction can therefore improve the probability vector while still moving a low-margin sample across a decision boundary. In this setting, probability redistribution may improve NLL but can also create correct-to-wrong transitions that are invisible if only distributional fit is inspected.

The comparison with DC illustrates this distinction. DC already uses regularized full affine calibration and validation-NLL selection, and on the benchmark SN setting StaCOR and DC are not reliably separated in NLL. Under benchmark IDN, DC has a small but stable NLL advantage, but StaCOR still improves the decision-stability metrics in the

paired comparison. Thus, the appropriate interpretation is not that StaCOR dominates DC. Rather, DC can provide slightly stronger probability optimization in some settings, whereas StaCOR shifts the trade-off toward preserving baseline-correct decisions. This pattern supports the interpretation that parameter regularization alone does not impose sample-specific preservation of decisions that clean evidence verifies as correct.

The external-validation results further sharpen this point. Across the frozen external datasets, DC has higher pooled Macro-F1 than StaCOR under both SN and IDN, while StaCOR retains lower harmful flips and higher retention. This pattern reinforces that decision preservation is not equivalent to maximizing Macro-F1. Macro-F1 summarizes final classification performance across classes, while retention asks whether the correction avoids regressing on decisions that were already correct for the frozen baseline. Both quantities are useful, but they answer different deployment questions.

The VBLL-full MC comparison suggests a complementary lesson. More flexible full class coupling does not automatically yield a better probability–decision operating point. The observed results are consistent with the possibility that excessive cross-class adjustment contributes to an unfavorable probability–decision trade-off in this setting. StaCOR should therefore be interpreted as a method for choosing a particular probability–decision trade-off, not as a method whose primary claim is to minimize NLL in every comparison.

5.2. Interpreting the Two StaCOR Components

The two controls in StaCOR operate at different levels. Regularized Class Coupling is a parameter-level capacity control. It allows the correction head to use cross-class information, which can be more expressive than purely diagonal calibration, while off-diagonal shrinkage discourages excessive class-to-class adjustments. In this sense, it provides an intermediate regularization regime between independent per-class rescaling and unrestricted full coupling. The empirical results are consistent with this design choice in the evaluated setting, but they should not be read as establishing a universal coupling structure.

Selective Decision Stability is instead a sample-level boundary-violation penalty. It acts only on clean head-training samples for which the frozen noisy-support baseline is already correct, and it is reactive: it penalizes corrected logits only when such a sample crosses the decision boundary. It does not predict whether a future query correction will be harmful, and it is not an oracle or a detector of harmful flips. RCC controls the global parameter-level flexibility of the correction map, whereas SDS imposes a local sample-level penalty when a verified baseline-correct decision crosses the classification boundary.

The $\beta = 1$ versus $\beta = 0$ ablation is consistent with this interpretation. Across the development benchmarks, the final objective improves Macro-F1, reduces harmful flip, and improves retention in every dataset–noise cell; the frozen external-validation results show the same directional pattern for the reported decision metrics. At the same time, the stability term can incur a small probability–quality cost. The ablation shows small measurable NLL costs in some pooled and dataset-level settings. This supports the view that Selective Decision Stability trades a small amount of probabilistic fit for more consistent preservation of baseline-correct decisions.

The frozen Selective-KL comparator provides a narrower objective-level check. Its scale-matched results favor SDS on Macro-F1, harmful flip, and retention in all four reported scopes, while its NLL comparison is significant only under development SN and external SN. This supports the stated SDS trade-off under the evaluated protocol, not a general ranking of all decision-preserving objectives.

This conservatism can also suppress some beneficial changes. Because the stability term prioritizes baseline-correct samples, it reduces the freedom of the correction head after those samples cross decision boundaries. That asymmetry is intentional: StaCOR

prioritizes avoiding regressions on predictions that were already correct. Regularized Class Coupling and Selective Decision Stability are therefore not redundant. The two controls are designed to provide complementary forms of control: global regularization of cross-class correction and local penalties for baseline-correct boundary violations.

5.3. Applicability Under Fixed-Context Constraints

StaCOR is intended for deployments in which the original TabPFN context is assumed to remain fixed, cached, version-locked, shared, audited, or costly to refresh during the serving cycle. This is an operational rather than informational boundary: clean calibration evidence can still be available to an external adaptation pipeline. The boundary is important because the clean-label usage experiments show that, when clean samples can be inserted directly into the TabPFN context, direct context utilization generally remains preferable. Under the same clean-label budget, Direct Append usually provides the strongest NLL and Macro-F1 performance, while Direct Replace also generally improves these two metrics over output-side correction despite keeping the context size fixed. The advantage of Direct Replace indicates that the direct-context benefit cannot be explained only by a larger context; the experiments indicate that direct incorporation of clean labels is generally more effective for NLL and Macro-F1.

This does not make the direct-context policies StaCOR variants or weak baselines. They represent a different deployment regime in which the serving context can be changed. In such settings, using clean labels directly in the context is usually the more effective strategy. StaCOR is instead designed for cases where that option is operationally restricted: the support set may be cached for reproducibility, locked for auditability, shared across a serving system, or expensive to rebuild and validate. In those cases, output-side refinement provides a way to use auxiliary clean evidence while leaving the original noisy context unchanged.

The serving results support this deployment distinction. StaCOR adds only a small correction-head overhead relative to the measured total latency in the evaluated workloads, while preserving the original context size. Direct Append, by contrast, necessarily increases the serving context as the clean-label budget grows. Table 7 and Figure 5 show that this context expansion consistently changes memory and context requirements, whereas the latency effect varies substantially across workloads. Thus, the serving conclusion is not that appending clean samples always imposes a large latency penalty, but that its resource impact is workload- and budget-dependent.

These results define StaCOR's intended use case rather than establishing direct context utilization as an inferior strategy. When context modification is feasible and operationally acceptable, direct clean-label utilization should generally be preferred. When the original TabPFN context must remain fixed or is inconvenient to refresh, StaCOR offers a lightweight output-side alternative with an explicit decision-stability objective.

5.4. Limitations and Future Directions

Several limitations follow directly from the experimental evidence. One limitation is that StaCOR does not minimize NLL in all comparisons. In particular, DC has a small but stable NLL advantage under benchmark IDN, and it also has favorable Macro-F1 on the frozen external-validation datasets. This limitation is consistent with the method's positioning: StaCOR is designed to improve the probability–decision trade-off, especially harmful-flip reduction and retention, rather than to serve as a pure NLL optimizer.

Second, StaCOR is not the preferred strategy when clean labels can freely be incorporated into the TabPFN context. The clean-label usage experiments indicate that Direct Append and Direct Replace generally make better use of the same clean-label budget. This

constrains the deployment scope of StaCOR to fixed-context settings. It also suggests a useful direction for future work: output-side correction and context-refresh policies could be studied as complementary tools rather than mutually exclusive alternatives.

Third, Selective Decision Stability is intentionally conservative. By protecting baseline-correct decisions, it can reduce harmful changes, but it can also suppress potentially beneficial corrections near the decision boundary. The ablation further shows that decision preservation carries a small measurable NLL cost in some settings. Future work could investigate adaptive stability strength based on decision geometry, calibration evidence, or deployment-specific risk tolerance, with the goal of providing finer Pareto control between probability fit and decision preservation.

Fourth, the evaluation scope remains limited. The formal study covers seven datasets: four development benchmarks and three frozen external-validation datasets. These datasets provide useful diversity, but they do not cover all tabular domains or all deployment regimes. Sensorless Drive Diagnosis was also excluded from the formal evaluation because the frozen TabPFN configuration used in this study did not support the 11-class Sensorless task under the adopted evaluation pipeline. Broader multi-class regimes and larger benchmark suites remain important future validation targets.

Finally, the label-noise protocols are controlled synthetic perturbations. SN and IDN are useful for repeatable stress testing, but real annotation errors may involve richer forms of instance dependence, annotator effects, structured corruption, or temporal variation that are not fully represented by the controlled protocols used here. StaCOR also requires additional auxiliary clean evidence, so it is not applicable when no trusted labels are available after deployment. Future work should therefore examine broader real-world noise processes, larger and more diverse tabular benchmarks, and integrations between output-side stability objectives and practical context-management policies.

6. Conclusions

Noisy support labels can degrade TabPFN predictions, and output-side calibration can improve posterior quality while disturbing hard decisions that were already correct. This paper introduced StaCOR, a selectively constrained fixed-context output refinement method that leaves the noisy TabPFN support set unchanged and trains a lightweight correction head on auxiliary clean evidence. The method combines Regularized Class Coupling, which controls cross-class correction capacity, with Selective Decision Stability, which discourages boundary-crossing regressions on baseline-correct clean training examples.

Across four development benchmarks and three frozen external-validation datasets under SN and IDN protocols, StaCOR provides a competitive probability–decision trade-off. Its clearest contribution is the joint design of parameter-level cross-class capacity control and sample-level selective protection of baseline-correct decisions in fixed-context output refinement. The stability ablation and frozen Selective-KL comparator support the role of SDS in this evaluated trade-off, while also showing that decision preservation can carry a small NLL cost in some settings.

The results also define the method’s boundary. When clean samples can be inserted into or used to replace the TabPFN context, direct clean-label utilization generally remains preferable. StaCOR is most relevant when the original served context is operationally fixed, cached, version-locked, shared, audited, or costly to refresh. More broadly, the study suggests that posterior correction for noisy-support TabPFN deployments should be evaluated jointly in terms of probability quality and decision stability.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/electronics15204784/s1>, Section S1: detailed formal protocol and

clean-label budget; Section S2: frozen Selective-KL comparator; Section S3: seed-clustered bootstrap sensitivity. Table S1: Frozen formal protocol and clean-label budget details. Support/query values are caps except where actual values are stated. Table S2: Frozen dataset-specific Selective-KL scale factors α_d , taken directly from the frozen normalization configuration. Table S3: Frozen three-way aggregate results at $\eta > 0$, with equal weighting over eta cells and fixed datasets. Lower is better for NLL and Harmful; higher is better for Macro-F1 and Retention. Table S4: Paired bootstrap differences, StaCOR (SDS) minus Selective-KL, from the frozen C5 comparison. Negative favors SDS for NLL/Harmful; positive favors SDS for Macro-F1/Retention. Table S5: Episode-level versus seed-clustered paired bootstrap sensitivity. Differences are SDS minus the reference method; negative favors SDS for NLL/Harmful and positive favors SDS for Macro-F1/Retention.

Author Contributions: Conceptualization, Y.H.; methodology, Y.H.; software, Y.H.; validation, Y.H.; formal analysis, Y.H.; investigation, Y.H.; data curation, Y.H.; visualization, Y.H.; writing—original draft preparation, Y.H.; writing—review and editing, Y.H., M.S., L.K. and K.S.; supervision, K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available from the original sources cited in the manuscript. The code used to reproduce the experiments will be made publicly available in an online repository upon publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DC	Dirichlet calibration.
ECE	Expected calibration error.
IDN	Instance-dependent noise.
NLL	Negative log-likelihood.
SN	Symmetric noise.
StaCOR	Stability-aware Coupling-Regularized Output Refinement
TabPFN	Tabular Prior-Data Fitted Network.

References

- Hollmann, N.; Müller, S.; Eggenberger, K.; Hutter, F. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *Proceedings of the International Conference on Learning Representations*; ICLR: Appleton, WI, USA, 2023.
- Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S.B.; Schirmer, R.T.; Hutter, F. Accurate Predictions on Small Data with a Tabular Foundation Model. *Nature* **2025**, *637*, 319–326. <https://doi.org/10.1038/s41586-024-08328-6>.
- Frenay, B.; Verleysen, M. Classification in the Presence of Label Noise: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* **2014**, *25*, 845–869. <https://doi.org/10.1109/TNNLS.2013.2292894>.
- Xia, X.; Liu, T.; Han, B.; Wang, N.; Gong, M.; Liu, H.; Niu, G.; Tao, D.; Sugiyama, M. Part-dependent Label Noise: Towards Instance-dependent Label Noise. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2020; Volume 33, pp. 7597–7610.
- Cheng, H.; Zhu, Z.; Li, X.; Gong, Y.; Sun, X.; Liu, Y. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In *Proceedings of the International Conference on Learning Representations*; ICLR: Appleton, WI, USA, 2021.
- Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
- Kull, M.; Perello Nieto, M.; Kängsepp, M.; Silva Filho, T.; Song, H.; Flach, P. Beyond Temperature Scaling: Obtaining Well-Calibrated Multi-Class Probabilities with Dirichlet Calibration. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2019; Volume 32.

8. Harrison, J.; Willes, J.; Snoek, J. Variational Bayesian Last Layers. In *Proceedings of the The Twelfth International Conference on Learning Representations*; ICLR: Appleton, WI, USA, 2024.
9. Zhang, J.; Kailkhura, B.; Han, T.Y.J. Mix-n-Match: Ensemble and Compositional Methods for Uncertainty Calibration in Deep Learning. In *Proceedings of the 37th International Conference on Machine Learning*, Virtual Event, 13–18 July 2020; pp. 11117–11128.
10. Rahimi, A.; Shaban, A.; Cheng, C.A.; Hartley, R.; Boots, B. Intra Order-Preserving Functions for Calibration of Multi-Class Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2020; Volume 33, pp. 13456–13467.
11. Ma, X.; Blaschko, M.B. Meta-Cal: Well-Controlled Post-Hoc Calibration by Ranking. In *Proceedings of the 38th International Conference on Machine Learning*, Virtual Event, 18–24 July 2021; pp. 7235–7245.
12. Qu, J.; Holzmüller, D.; Varoquaux, G.; Le Morvan, M. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *Proceedings of the 42nd International Conference on Machine Learning*, Vancouver, BC, Canada, 13–19 July 2025; pp. 50817–50847.
13. Arazi, A.; Shapira, E.; Reichart, R. TabSTAR: A Tabular Foundation Model for Tabular Data with Text Fields. In *Proceedings of the Advances in Neural Information Processing Systems 38*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2025; pp. 190695–190748. <https://doi.org/10.52202/085713-5726>.
14. Arbel, M.; Salinas, D.; Hutter, F. EquiTabPFN: A Target-Permutation Equivariant Prior Fitted Network. In *Proceedings of the Advances in Neural Information Processing Systems 38*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2025; pp. 69661–69684. <https://doi.org/10.52202/085713-2097>.
15. García, P.; de Curtò, J.; de Zarzà, I.; Cano, J.C.; Calafate, C.T. Foundation Models for Cybersecurity: A Comprehensive Multi-Modal Evaluation of TabPFN and TabICL for Tabular Intrusion Detection. *Electronics* **2025**, *14*, 3792. <https://doi.org/10.3390/electronics14193792>.
16. Nawaz, A.; Ahmad, A.; Khan, S.S. Assessing the Robustness of Tabular Prior-Data Fitted Network Classifier. In *Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data*; ICML: Vancouver, BC, Canada, 2025.
17. Patrini, G.; Rozza, A.; Menon, A.K.; Nock, R.; Qu, L. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017.
18. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2018; Volume 31.
19. Zong, C.C.; Wang, Y.W.; Xie, M.K.; Huang, S.J. Dirichlet-Based Prediction Calibration for Learning with Noisy Labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Advancement of Artificial Intelligence (AAAI): Washington, DC, USA, 2024; Volume 38, pp. 17254–17262. <https://doi.org/10.1609/aaai.v38i15.29672>.
20. Wang, Z.; Zhou, M.; Zhao, Y.; Zhang, F.; Wang, J.; Qian, B.; Liu, Z.; Ma, P.; Ma, Q. Electrical Power Edge-End Interaction Modeling with Time Series Label Noise Learning. *Electronics* **2023**, *12*, 3987. <https://doi.org/10.3390/electronics12183987>.
21. Tomani, C.; Cremers, D.; Buettner, F. Parameterized Temperature Scaling for Boosting the Expressive Power in Post-Hoc Uncertainty Calibration. In *Computer Vision—ECCV 2022*; Springer Nature: Cham, Switzerland, 2022; pp. 555–569. https://doi.org/10.1007/978-3-031-19778-9_32.
22. Silva Filho, T.; Song, H.; Perello-Nieto, M.; Santos-Rodriguez, R.; Kull, M.; Flach, P. Classifier Calibration: A Survey on How to Assess and Improve Predicted Class Probabilities. *Mach. Learn.* **2023**, *112*, 3211–3260. <https://doi.org/10.1007/s10994-023-06336-7>.
23. Esaki, Y.; Nakamura, A.; Kawano, K.; Tokuhisa, R.; Kutsuna, T. Accuracy-Preserving Calibration via Statistical Modeling on Probability Simplex. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, Valencia, Spain, 2–4 May 2024; pp. 1666–1674.
24. Zhang, Y.; Batista, G.E.; Kanhere, S.S. Instance-Wise Monotonic Calibration by Constrained Transformation. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, Rio de Janeiro, Brazil, 21–25 July 2025; pp. 4920–4932.
25. Cao, Z.; Li, Y.; Kim, D.H.; Shin, B.S. Deep Neural Network Confidence Calibration from Stochastic Weight Averaging. *Electronics* **2024**, *13*, 503. <https://doi.org/10.3390/electronics13030503>.
26. Li, X.; Fei, J.; Xie, J.; Li, D.; Jiang, H.; Wang, R.; Qi, Z. Open Set Recognition for Malware Traffic via Predictive Uncertainty. *Electronics* **2023**, *12*, 323. <https://doi.org/10.3390/electronics12020323>.
27. Pan, Q.; Meng, Z. Hybrid Uncertainty Calibration for Multimodal Sentiment Analysis. *Electronics* **2024**, *13*, 662. <https://doi.org/10.3390/electronics13030662>.
28. Nakai, K. *Yeast*; UCI Machine Learning Repository: Irvine, CA, USA, 1991. <https://doi.org/10.24432/C5KG68>.
29. Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J. *Wine Quality*; UCI Machine Learning Repository: Irvine, CA, USA, 2009. <https://doi.org/10.24432/C56S3T>.
30. Malerba, D. *Page Blocks Classification*; UCI Machine Learning Repository: Irvine, CA, USA, 1994. <https://doi.org/10.24432/C5J590>.

31. Alpaydin, E.; Kaynak, C. *Optical Recognition of Handwritten Digits*; UCI Machine Learning Repository: Irvine, CA, USA, 1998. <https://doi.org/10.24432/C50P49>.
32. Vergara, A. *Gas Sensor Array Drift at Different Concentrations*; UCI Machine Learning Repository: Irvine, CA, USA, 2012. <https://doi.org/10.24432/C5MK6M>.
33. UCI Machine Learning Repository. *Crop Mapping Using Fused Optical-Radar Data Set*; UCI Machine Learning Repository: Irvine, CA, USA, 2020. <https://doi.org/10.24432/C5G89D>.
34. Srinivasan, A. *Statlog (Landsat Satellite)*; UCI Machine Learning Repository: Irvine, CA, USA, 1993. <https://doi.org/10.24432/C55887>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.