

Article

When Feature Alignment Helps, Hurts, or Washes out: Carrier-Dependent Domain Adaptation for Cherenkov Telescope Event Classification

Antonio Pagliaro ^{1,2,3} 

¹ INAF—Istituto di Astrofisica Spaziale e Fisica Cosmica di Palermo, Via Ugo La Malfa 153, I-90146 Palermo, Italy; antonio.pagliaro@inaf.it

² Istituto Nazionale di Fisica Nucleare Sezione di Catania, Via Santa Sofia, 64, I-95123 Catania, Italy

³ ICSC—Centro Nazionale di Ricerca in HPC, Big Data e Quantum Computing, I-40121 Bologna, Italy

Abstract

Feature alignment domain adaptation (DA) methods such as CORAL and MMD are widely used to mitigate distribution shift between training (source) and deployment (target) domains. They are typically evaluated on a single downstream classifier, which implicitly assumes that the alignment benefit transfers across model families. We test this assumption on a controlled class-symmetric shift of Imaging Atmospheric Cherenkov Telescope features, with five carrier classifiers, three alignment methods, and fifteen seeds per cell. Establishing this requires first correcting an evaluation protocol that we believe to be common in benchmarks built by perturbing a single sample: when the target consist of a perturbed copy of data that the carrier was partly trained on, the measured benefit of adaptation is inflated by a factor that tracks the carrier's capacity to memorise (from 1.0 for logistic regression to 5.7 for an unbounded-depth random forest), so that such a benchmark reports a carrier gradient whether or not one actually exists. When scored only on events that no carrier has seen, the carrier still matters (indeed, twice over): the same shift costs a hyperplane carrier 0.004 of AUC and a tree ensemble 0.006–0.008, and CORAL then recovers 58% of that loss on logistic regression, against 84–98% on the tree ensembles, a difference that RBF-kernel MMD does not show on any carrier for which the ratio is determined. A measurement of marginal sensitivity taken without any adaptation reproduces the ordering. Two effects that would be reported by a full-target protocol are absent on held-out events: a benefit growing with tree depth, and an advantage of covariance alignment over mean matching on deep trees. We propose a two-axis framework (shift symmetry $\times$ carrier marginal sensitivity), then test its class-symmetric row and give practical recommendations for evaluating and deploying domain adaptation under shift.



Academic Editors: Jong-Ha Lee and Sitara Afzal

Received: 17 September 2026

Revised: 5 October 2026

Accepted: 7 October 2026

Published: 9 October 2026

Copyright: © 2026 by the author.

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: domain adaptation; distribution shift; feature alignment; CORAL; classifier sensitivity; tree-based classifiers; benchmark design; evaluation protocol; interpretable machine learning; gamma ray astronomy

1. Introduction

Supervised classifiers deployed outside their training distribution suffer predictable performance degradation. This *distribution shift* arises whenever the source domain (training data) and target domain (deployment data) differ in their feature distributions, label

distributions, or both. The problem of distribution shift is pervasive in science and engineering; for example, Monte Carlo simulations used for training differ systematically from real observations [1,2], sensor calibrations drift between laboratory and field conditions, and population demographics shift between clinical trials and deployment cohorts [3].

Domain adaptation (DA) addresses this problem by aligning the source and target distributions before or during classification. Feature alignment methods such as CORAL [4], MMD-based alignment [5], and simple mean matching transform source features to match the target's marginal distribution, under the assumption that aligned features will yield better cross-domain classification [6].

A critical but undercharacterised variable in this pipeline is the *carrier classifier*, which is the model that receives the (possibly aligned) features and produces predictions. The domain alignment literature predominantly evaluates alignment methods on a single carrier per benchmark, typically a neural network for image tasks or logistic regression for tabular tasks. The implicit assumption is that the benefit of alignment will transfer across classifiers, e.g., if CORAL helps a logistic regression, then it should help a random forest by a comparable amount.

However, this assumption does not hold. In this paper, we show that the effectiveness of feature alignment DA depends on the carrier classifier's structural sensitivity to marginal feature distributions. In fact, this happens twice: first in how much a class-symmetric shift damages each carrier, and again in how much of that damage is repaired by alignment. On a controlled benchmark with five classifiers spanning three structural families (linear, neural, and tree-based), the same shift costs a hyperplane carrier 0.004 of AUC and a threshold-based ensemble 0.006 to 0.008, between 1.4 and 1.7 times as much; CORAL then recovers 58% of the loss on logistic regression, against 84–98% on the tree ensembles.

Establishing this required correcting an evaluation protocol that we believe is common in benchmarks following this construction, the effect of which is larger than the effect under study. When the target domain is obtained by perturbing a sample that includes data on which the carrier was trained, most of the “target” will consist of perturbed copies of training events. A high-capacity carrier memorises these events, and the measured benefit of adaptation is then inflated by a factor that tracks memorisation capacity, reaching 5.7 for an unbounded-depth random forest while remaining at 1.0 for logistic regression. Reporting a benchmark in this way results in a carrier gradient being manufactured out of a capacity gradient. Therefore, all of the results below are computed on target events that no carrier has seen in any form.

We propose a two-axis framework for reasoning about DA effectiveness:

1. **Shift symmetry:** Whether the source–target shift affects both classes equally (class-symmetric) or differentially (class-asymmetric).
2. **Carrier marginal sensitivity:** Whether the classifier's decision boundaries depend on absolute feature locations (marginal-sensitive, e.g., tree-based) or only on relative geometry (marginal-robust, e.g., linear).

Under class-symmetric shift, feature alignment yields a larger gain on marginal-sensitive carriers. On marginal-robust ones, it is still helpful, but less so; this is because the translational part of the shift is invisible to their ranking, and only the remainder is left to repair. Whatever such a carrier loses at a fixed decision threshold is recovered by re-selecting the threshold, rather than by aligning the features. We empirically populate the class-symmetric row of this framework; the class-asymmetric row remains a testable prediction for future work on real-data sim-to-real scenarios.

We provide the following contributions:

- A measurement of how much a class-symmetric location shift costs each carrier and how much of that cost feature alignment gives back. Threshold-based carriers lose 1.4

to 1.7 times as much AUC as a hyperplane carrier; CORAL recovers 84–98% of this, against 58% for logistic regression, so the absolute gains differ by a factor of about 2.5. The recovered fraction is specific to the alignment method. Under RBF-kernel MMD, the carriers still differ in damage but recover 0.85 to 0.91 on every carrier for which the ratio is determined, including logistic regression.

- A measure of carrier marginal sensitivity that uses neither adaptation nor target data, so that the ordering of carriers is established independently of the effect that it is meant to explain.
- A two-axis conceptual framework (shift symmetry $\times$ carrier marginal sensitivity) that predicts when feature alignment DA helps, hurts, or washes out. Only its class-symmetric row is tested here.
- An evaluation protocol finding that we believe generalizes well beyond this benchmark, namely, that when the target domain is a perturbed copy of a sample the carrier was partly trained on, the measured benefit of adaptation is inflated by a factor that tracks the carrier's capacity to memorise. This ranges from 1.0 for logistic regression to 5.7 for an unbounded-depth random forest. Benchmarks following this construction will consequently report a carrier gradient even where none exists.
- An ablation over shift type and intensity, showing that the carrier effect is a property of location shifts. The effect scales with shift intensity, is reproduced by a pure location shift, and vanishes under a pure noise shift.
- A controlled synthetic benchmark with physically motivated perturbations, designed for reproducibility and released as open-source code.

The rest of this paper is organised as follows: Section 2 introduces the DA methods and classifier families used in the study; Section 3 describes the benchmark design, including the synthetic perturbation framework and evaluation protocol; Section 4 presents the main results; Section 5 interprets the findings, discusses limitations, and offers practical recommendations; finally, Section 6 concludes the paper.

2. Background

2.1. Domain Adaptation via Feature Alignment

Domain adaptation (DA) addresses the problem of deploying a classifier trained on a source distribution $P_s(\mathbf{x}, y)$ to a target distribution $P_t(\mathbf{x}, y)$ where $P_s \neq P_t$. In the unsupervised setting, labelled data are available only in the source domain; the target domain provides unlabelled samples from $P_t(\mathbf{x})$. Kouw and Loog reviewed the field in [7]; the methods studied here belong to its feature-based branch.

Feature alignment methods transform the source features so that their distribution matches the target. The core assumption is that if the marginal feature distributions are aligned ($P_s(\mathbf{x}) \approx P_t(\mathbf{x})$), a classifier trained on the aligned source will generalise to the target. This assumption is formalised by Ben-David et al. [6], who bounded the target error in terms of the source error plus the distributional divergence between domains, quantified by the $\mathcal{H}\Delta\mathcal{H}$ -distance. The proxy A-distance used here estimates the related $\mathcal{H}$ -divergence, obtained by training a discriminator to separate the two domains, which is the quantity accessible without labels.

2.2. CORAL

Correlation alignment (CORAL) [4] aligns the second-order statistics (covariance) of source and target features. Given source features $\mathbf{X}_s$ with covariance $\mathbf{C}_s$ and target features $\mathbf{X}_t$ with covariance $\mathbf{C}_t$, the CORAL transform is as follows:

$$\hat{\mathbf{X}}_s = (\mathbf{X}_s - \boldsymbol{\mu}_s) \mathbf{C}_s^{-1/2} \mathbf{C}_t^{1/2} + \boldsymbol{\mu}_t \quad (1)$$

where μ_s, μ_t are the domain means. A regularisation term $\lambda \mathbf{I}$ is added to the covariance matrices to ensure numerical stability. CORAL aligns both the mean and the covariance of the source to the target.

2.3. MMD-Based Alignment

The maximum mean discrepancy (MMD) [5] measures the distance between two distributions in a reproducing kernel Hilbert space. A linear projection $\mathbf{W}$ can be learned to minimise the empirical MMD between $\mathbf{W}\mathbf{X}_s$ and $\mathbf{X}_t$, with regularisation $\lambda \|\mathbf{W} - \mathbf{I}\|_F^2$ to prevent trivial collapse. The kernel determines which statistics are matched: with a linear kernel, the squared MMD reduces to the distance between the two mean embeddings, so it is a first-order criterion that, for an affine transform that already equalises the means, is minimised by mean matching itself. In closed form, CORAL provides second-order (covariance) alignment; matching higher-order structure requires a characteristic kernel such as the RBF. With an RBF kernel, the alignment can capture nonlinear distributional differences, though at the cost of a gradient-based optimisation.

2.4. Mean Matching

The simplest feature alignment method shifts the source mean to match the target: $\hat{\mathbf{X}}_s = \mathbf{X}_s - \mu_s + \mu_t$. This corrects the first-order (location) shift, but leaves the covariance unchanged. We include mean matching as a baseline in order to isolate the contribution of location alignment from that of covariance alignment.

2.5. Carrier Classifiers and Marginal Sensitivity

A DA method does not classify events directly; it transforms features before passing them to a carrier classifier. The DA literature predominantly evaluates alignment methods using a single carrier, commonly neural networks for image benchmarks or logistic regression for tabular ones. However, the interaction between the carrier's decision boundary structure and the alignment method is rarely studied.

We identify two structural families relevant to this interaction:

- **Hyperplane classifiers** (logistic regression, linear SVM): The score is a linear function of the features. A uniform translation of all features adds the same constant to every score, so the ranking of events is exactly preserved and any ranking metric is unchanged. The hyperplane itself, however, is displaced relative to the data, so a decision taken at a fixed score threshold is not preserved.
- **Threshold-split classifiers** (random forest [8], ExtraTrees [9], LightGBM [10]): The decision regions are unions of axis-aligned hyper-rectangles. Each split depends on the absolute value of a single feature, making the classifier sensitive to marginal shifts that displace the feature distribution relative to the learned threshold.

Neural networks (MLPs) occupy an intermediate position; with sufficient depth and nonlinearity, they can approximate both hyperplane and threshold-like decision boundaries, and their marginal sensitivity depends on the architecture and training dynamics.

3. Method

We design a controlled benchmark to isolate the interaction between domain adaptation methods and carrier classifiers under a known distribution shift. The benchmark uses physically motivated synthetic perturbations to construct a source–target pair with identical labels and a shift that is fully characterised by construction.

3.1. Synthetic Perturbation Framework

The source domain consists of Monte Carlo simulated events with ten tabular features describing geometric and photometric properties of particle shower images from Imaging Atmospheric Cherenkov Telescopes (IACTs) [11]. We apply five families of physically motivated perturbations at the feature level:

1. **Additive noise injection:** Poisson-like noise scaled by $\sqrt{|\text{SIZE}|}$, modelling night sky background variation. Secondary effects on concentration features and geometric broadening of image width and length.
2. **Multiplicative gain uncertainty:** Per-event Gaussian factor on SIZE ($\sigma = 5\%$ at headline intensity), modelling residual flat fielding errors.
3. **Global attenuation:** Deterministic multiplicative reduction of SIZE, WIDTH, and LENGTH, modelling atmospheric absorption (headline: 20% SIZE reduction).
4. **Zenith-dependent size shift:** Size-dependent attenuation with stronger effect at low SIZE, plus geometric elongation of LENGTH and broadening of WIDTH; this models oblique viewing at higher zenith angles.
5. **Threshold shift:** Selective rejection of events below a SIZE threshold, modelling effective energy threshold instability.

Each perturbation is parameterised by a single intensity scalar in $[0, \infty)$: intensity = 0 is the identity (no perturbation), and intensity = 1 corresponds to the headline level calibrated against published instrumental systematics. All perturbations are deterministic given a fixed random seed.

While the perturbation families are motivated by IACT physics, the framework is domain-general; any tabular classification task with known sources of distributional shift can instantiate the same design by replacing the perturbation families with domain-appropriate transformations.

3.2. Benchmark Scenario

The headline benchmark scenario (denoted S0) is constructed as follows:

- **Source:** 24,946 events (7895 gamma ray, 17,051 hadron) from the CTA Prod5 public Monte Carlo release [12]. Their production is described by Gueta [13], with quality cuts applied ($\text{SIZE} > 50$ photo-electrons, $\text{WIDTH} > 0$, $\text{LENGTH} > 0$).
- **Target:** The same events but with a composite perturbation applied, consisting of a combination of additive noise injection and global attenuation, both at intensity 1.0.
- **Preprocessing:** $\log_{10}(\text{SIZE})$ transform, followed by standardisation (fitted on source, applied to both domains). The log-transform addresses the power-law distribution of SIZE (coefficient of variation ≈ 8).

By construction, the source and target share identical labels. The shift is class-symmetric; both gamma and hadron populations are perturbed identically, so the class-conditional structure is transported identically for the two classes rather than being altered relative to one another. This design choice isolates the effect of marginal feature distribution shift from label shift or class-conditional shift.

Held-out target evaluation. Because the target is a perturbed copy of the same events, a carrier trained on 70% of the source would be scored partly on perturbed copies of its own training events if evaluated on the whole target. High-capacity carriers memorise these events, so the resulting figures measure memorisation as much as adaptation. An unbounded-depth random forest reaches AUC 0.909 on the full target, but 0.759 on the perturbed copies of events it has never seen. Every number reported in this paper is computed on the held-out target, that is, on the perturbed copies of the 30% of events excluded from training, which the carrier has not seen in any form. The alignment transforms are still

fitted on the full unlabelled source and target feature sets, as a practitioner would fit them, since that step uses no labels and no held-out information. We report this protocol because the distinction is easy to overlook in benchmarks built by perturbing a single sample and because it changes the reported benefit of domain adaptation by a factor of several times (Section 4.1).

Damage and recovery. With a held-out target, two quantities are well defined for each carrier c . The *shift damage*

$$D_c = \text{AUC}_c(\text{held-out source}) - \text{AUC}_c(\text{held-out target}), \quad (2)$$

measured with the unadapted model, is the performance cost of the shift. The *recovery fraction* of an adaptation method m ,

$$R_{c,m} = \frac{\text{AUC}_{c,m}(\text{held-out target}) - \text{AUC}_c(\text{held-out target})}{D_c}, \quad (3)$$

is the share of that cost that the method then gives back. Here, R (unlike raw ΔAUC) is dimensionless, making it comparable across carriers with differing absolute accuracy.

D and the numerator of R are paired differences and, like every other difference in this paper, are summarised by their mean over seeds with a 95% t -interval. R itself is a ratio of two small quantities, and the per-seed ratio is unusable when the denominator is close to zero, as it is for the MLP. Therefore, we report R as a ratio of means, $\bar{\text{gain}}/\bar{D}$, with a percentile bootstrap interval. The fifteen carrier–seed pairs are resampled with replacement $B = 10^4$ times, the ratio of means is recomputed on each resample, and the interval is the 2.5th to 97.5th percentile of that distribution. Where the resulting interval contains zero, as it does for the MLP, we treat R as undetermined in the main results; for completeness, the point estimates and intervals for every cell are listed in Section 4.2.

3.3. Carrier Classifiers

We evaluate five classifiers spanning three structural families:

- **Linear:** Logistic regression (max_iter = 1000).
- **Tree-based:** LightGBM [10] ($n_{\text{est}} = 300$, max_depth = 6, learning rate = 0.05), random forest [8] ($n_{\text{est}} = 100$, max_depth = ∞ , criterion = entropy), ExtraTrees [9] (same hyperparameters as RF).
- **Neural:** MLP-shallow (hidden layers (64, 32), ReLU, Adam, early stopping).

All carriers are trained on (possibly aligned) source features and evaluated on target features. Determinism is ensured by fixing random seeds and setting $n_{\text{jobs}} = 1$ for tree-based classifiers.

3.4. Domain Adaptation Methods

We benchmark three feature alignment methods plus a no-adaptation baseline. All three share the same affine form $\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu}_s) \mathbf{W} + \boldsymbol{\mu}_t$, applied to the standardised source features, and differ only in the matrix $\mathbf{W}$:

- **Source-only** (src-only): No adaptation; train on source, test on target.
- **Mean matching:** $\mathbf{W} = \mathbf{I}$; shifts the source mean onto the target mean, while the covariance is unchanged.
- **CORAL** [4]: $\mathbf{W} = (\mathbf{C}_s + \lambda \mathbf{I})^{-1/2}(\mathbf{C}_t + \lambda \mathbf{I})^{1/2}$, with $\mathbf{C}_s$, $\mathbf{C}_t$ as the sample covariances and $\lambda = 10^{-3}$. This whitens the source covariance and re-colours it with the target covariance, so that both mean and covariance match. The matrix square roots are computed by eigendecomposition, with eigenvalues clipped at 10^{-12} .

- **MMD-rbf:** $\mathbf{W}$ is obtained by minimising the empirical squared maximum mean discrepancy [5] between the transformed source and the target under an RBF kernel (median-heuristic bandwidth on 500 source points), plus the penalty $\lambda \|\mathbf{W} - \mathbf{I}\|_F^2$ with $\lambda = 10^{-3}$, by L-BFGS-B from $\mathbf{W} = \mathbf{I}$ for at most 100 iterations on 500-point subsamples.

Why is there no “MMD-linear” method? A fourth method, MMD with a linear kernel, is absent because it is not a distinct method within this family. Under the linear kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$, the squared MMD equals $\|\mathbb{E}[\hat{\mathbf{x}}] - \mathbb{E}[\mathbf{x}_t]\|^2$, which is the distance between the first-order mean embeddings. Within the affine family above, this quantity is zero for every $\mathbf{W}$. Because the transform already moves the source mean onto the target mean, the penalised objective is minimised at $\mathbf{W} = \mathbf{I}$, which is mean matching. Thus, the two closed-form extremes of the family, mean matching and CORAL, are its informative members, and MMD with a linear kernel would add nothing to either.

All adaptation methods are fitted on the full source and target feature sets (unsupervised; target labels are not used for alignment). The aligned source features are then used to train the carrier classifier.

3.5. An Independent Measure of Marginal Sensitivity

The carrier gradient in Section 4.2 ranks the five carriers using a structural argument (hyperplanes versus threshold splits). To avoid circularity, we also measure the marginal sensitivity using a procedure that involves no domain adaptation and no target data. For each carrier and seed, the source-only model trained on the 70% source split is evaluated on the 30% held-out source split after every standardised feature has been translated by a common offset $\pm\delta$, then record the AUC degradation:

$$S(\delta) = \text{AUC}_0 - \frac{1}{2} [\text{AUC}(+\delta) + \text{AUC}(-\delta)] \quad (4)$$

where AUC_0 is the held-out AUC, without translation, for $\delta \in \{0.1, 0.25, 0.5, 1.0\}$. A uniform translation is the simplest class-symmetric location shift, leaving the ranking produced by any linear score exactly unchanged; thus, $S(\delta) = 0$ for logistic regression by construction, and it crosses the axis-aligned thresholds of a tree in proportion to how densely the tree has placed them. $S(\delta)$ is threshold-free. We also record the fraction (at probability 0.5) of hard predictions that flip under the same translation, which is threshold-dependent and is reported separately because its different behaviour. The index $S(0.25)$ is then compared with the CORAL gain ΔAUC of the same carrier and seed.

3.6. Operating Points for the Q-Factor

The Q-factor of the main matrix is evaluated at the default operating point of predicted probability ≥ 0.5 . Because the score distribution of a carrier changes under shift and under alignment, a fixed threshold can exaggerate or hide a gain; therefore, we also evaluate the following for source-only, mean matching, and CORAL on every carrier: (i) Q on the target over a grid of thresholds from 0.05 to 0.95; (ii) Q on the target at the threshold θ^* that maximises Q on the 30% source validation split, transferred unchanged to the target, which is the operating point that a practitioner without target labels can actually choose; and (iii) the oracle maximum of Q over the grid on the target itself. Thresholds accepting fewer than 50 hadrons are excluded to keep Q finite.

3.7. Shift Diagnostics

We quantify the source–target shift independently of any DA method using three diagnostics computed in the preprocessed feature space:

- **Kernel MMD:** Empirical MMD with RBF kernel and median-heuristic bandwidth. Statistical significance assessed via permutation test ($B = 1000$, fixed seed for reproducibility).
- **Proxy A-distance [6]:** A logistic regression classifier is trained to distinguish source from target events; $\hat{d}_A = 2(1 - 2\epsilon)$, where ϵ is the 5-fold cross-validation error.
- **Calibration drift:** The expected calibration error (ECE) with 15 bins is computed for a logistic regression trained on source labels, evaluated on a source held-out set and on the target. The difference ΔECE quantifies how much the calibration degrades under shift.

3.8. Evaluation Metrics

We report three metrics:

- **F1 macro:** The harmonic mean of per-class F1, robust to class imbalance.
- **AUC:** The area under the ROC curve, our primary ranking metric.
- **Q-factor:** $\epsilon_\gamma / \sqrt{\epsilon_h}$, a domain-specific figure of merit for gamma/hadron separation in IACT analysis, where ϵ_γ and ϵ_h are the gamma and hadron acceptance rates.

Each of the twenty cells (five carriers $\times$ four methods, the baseline included) is evaluated with fifteen independent random seeds. We report mean $\pm$ standard deviation per cell. Because all cells share the same seed set and the seed determines both the train/test split and the classifier initialisation, the ΔAUC (method minus src-only) is computed as a paired difference seed-by-seed; we report its mean, a 95% t -interval with $n - 1 = 14$ degrees of freedom, and the paired t -test p -value. The fifteen seeds are independent draws of the split and initialisation on a fixed base dataset, so the intervals quantify the sampling variability of the split and training, not the variability across datasets. The full processing order is as follows: (i) build the source and target from the base events; (ii) $\log_{10}$ -transform SIZE; (iii) fit the standardiser on the source and apply it to both domains; (iv) fit the alignment on the standardised source and target features (unsupervised); (v) train the carrier on the aligned source; and (vi) evaluate on the held-out target.

4. Results

We evaluate five carrier classifiers and three domain adaptation methods on the controlled S0 benchmark (Section 3.2), with fifteen seeds per cell. Every cell uses the same seed set, and the seed fixes the train/test split and the classifier initialisation; thus, each difference between methods is computed as a paired difference seed-by-seed. We report the paired mean with its 95% t -interval ($n = 15$, 14 degrees of freedom), and where relevant, the paired t -test p -value. All target-side quantities are computed on the held-out target defined in Section 3.2, based on the reasoning set out in the next subsection.

4.1. Contents of the Target Set

We begin with a methodological result, which determines how the other numbers in this paper should be read.

In a benchmark of this kind, the target domain is obtained by perturbing a sample. It is natural to score the adapted model on the whole perturbed sample. However, if the carrier was trained on part of that sample, then most of the “target” will consist of perturbed copies of its own training events. Table 1 shows the results of this issue. A source-only random forest, grown to unbounded depth in the configuration used here, separates its training events perfectly (AUC 1.000), but separates events it has never seen before at AUC 0.766. Logistic regression, which has no capacity to memorise, scores 0.689 and 0.686 on the same two sets. The other carriers lie in between, in proportion to their capacity: the memorisation gap is 0.003 for logistic regression, 0.019 for the MLP, 0.128 for LightGBM and 0.23–0.24 for the two ensembles with unbounded depth.

The consequence for domain adaptation is not a small correction; when scored on the full perturbed sample, the source-only random forest reaches AUC 0.909 rather than 0.759, and the gain for CORAL rises from +0.006 to +0.035. To locate where this excess comes from, we decompose the pooled AUC into the three populations of pairs that it counts: both events drawn from the training set, both from the held-out set, or one from each (Figure 1b). For logistic regression, the three nearly coincide (0.685, 0.682, and 0.683), so the pooled AUC measures essentially just one quantity. For the random forest, these results are 0.962, 0.759, and 0.880, while for ExtraTrees they are 0.993, 0.753, and 0.930. By construction, the middle entry of each triple is the held-out AUC of Table 2, and agrees with it; thus, the pooled figure is a blend of a near-perfect memorised ordering, a genuine generalisation near 0.76, and a cross term that is high because memorised events receive extreme scores that clearly separate them from the middling scores of unseen events.

Table 1. Source-only AUC of each carrier on four event sets (mean over 15 seeds), together with the figure obtained by scoring the whole perturbed sample. The gap between the first and third columns is due to memorisation, which is negligible for a linear carrier and complete for the unbounded-depth ensembles. The last column is the gain for CORAL measured on the full perturbed sample, then divided by the same gain measured on held-out events.

Carrier	Training Events		Held-Out Events		Memor. Gap	Full Target	Gain Inflation
	No Shift	Shifted	No Shift	Shifted			
Logistic regression	0.689	0.685	0.686	0.682	0.003	0.684	1.0×
MLP-shallow	0.781	0.777	0.762	0.758	0.019	0.771	1.4×
LightGBM	0.901	0.850	0.773	0.765	0.128	0.825	1.9×
ExtraTrees	1.000	0.993	0.761	0.753	0.239	0.945	3.2×
Random Forest	1.000	0.962	0.766	0.759	0.234	0.909	5.7×

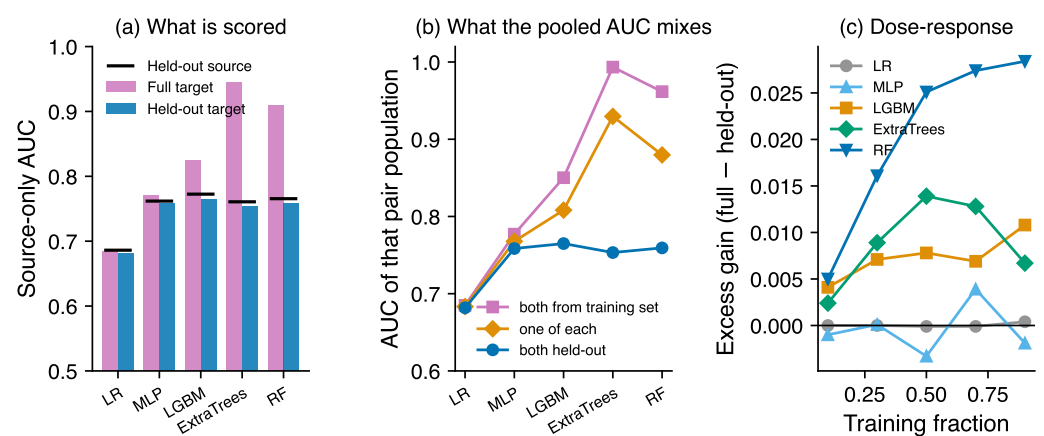


Figure 1. These figures show the evaluation protocol, what it does, and why: (a) shows the source-only AUC on the full perturbed sample and the held-out target, with the unshifted held-out source marked for reference; (b) shows the pooled AUC of the unadapted model decomposed into the three populations of pairs that it counts, which nearly coincide for a linear carrier but not for the ensembles; (c) shows the excess gain for CORAL consisting of the full target minus the held-out target, as a function of the fraction of the sample used for training. It can be seen that the excess gain is zero at every fraction for logistic regression, but rises and becomes saturated for the carriers with memorisation ability. The horizontal black line in (c) marks zero excess gain, that is, the value at which the full-target and held-out-target protocols agree.

The excess gain enters through different channels for different carriers, which is notable because a single mechanism does not account for all of them. For LightGBM the full-target gain is the same as the 70/30 mixture of the gains on the two subsets (residual +0.0001), so alignment helps more only on memorised events (+0.017 against +0.008). For ExtraTrees the subset gains are both small (+0.006 each), yet the full-target gain is +0.020; essentially all

of the excess sits in the cross term. for the random forest, both channels contribute (+0.031 on training events and a further +0.012 of cross-term residual). Despite these differences, the channels still share a dependence on the existence of a memorised population.

Table 2. S0 benchmark on the held-out target, 15 seeds per cell. Blocks, top to bottom: AUC; shift damage D (AUC lost by the unadapted carrier, held-out source versus held-out target) and its fraction R recovered by each method; paired Δ AUC relative to source-only, with the half-width of the paired 95% t -interval in parentheses (units of 10^{-3} ; an asterisk marks intervals containing zero); macro-F1 at the default operating point and its paired Δ ; and Q-factor at the default operating point. Here, R is a ratio of means with a bootstrap interval (Section 3.2), reported for every cell in Table 3; the MLP entries are undetermined because its damage is too small to divide by. The ranking metric and the two fixed-threshold metrics disagree on the linear carrier, where alignment changes the ranking only slightly but moves the score offset; however, Section 4.6 resolves this disagreement.

	LR	MLP	LightGBM	ExtraTrees	RF
<i>AUC</i>					
Source-only	0.682 $\pm$ 0.006	0.758 $\pm$ 0.007	0.765 $\pm$ 0.005	0.753 $\pm$ 0.005	0.759 $\pm$ 0.004
Mean matching	0.682 $\pm$ 0.006	0.760 $\pm$ 0.008	0.771 $\pm$ 0.005	0.760 $\pm$ 0.006	0.765 $\pm$ 0.006
CORAL	0.684 $\pm$ 0.006	0.762 $\pm$ 0.008	0.772 $\pm$ 0.005	0.759 $\pm$ 0.006	0.765 $\pm$ 0.006
MMD-rbf	0.686 $\pm$ 0.006	0.761 $\pm$ 0.009	0.772 $\pm$ 0.005	0.760 $\pm$ 0.006	0.765 $\pm$ 0.005
<i>Shift damage D and recovered fraction R</i>					
Damage D	0.0044	0.0034	0.0076	0.0074	0.0062
R , Mean matching	0.00	undet.	0.76	0.92	0.86
R , CORAL	0.58	undet.	0.98	0.84	0.98
R , MMD-rbf	0.89	undet.	0.90	0.91	0.85
<i>Paired ΔAUC vs. source-only (95% CI half-width, $\times 10^{-3}$)</i>					
Mean matching	0.0000 (0.0) *	+0.0013 (3.1) *	+0.0058 (1.5)	+0.0068 (2.1)	+0.0053 (1.8)
CORAL	+0.0025 (0.3)	+0.0040 (5.2) *	+0.0075 (1.6)	+0.0062 (1.6)	+0.0061 (2.1)
MMD-rbf	+0.0039 (0.5)	+0.0022 (4.9) *	+0.0068 (1.7)	+0.0067 (1.8)	+0.0053 (2.1)
<i>Macro-F1 at threshold 0.5</i>					
Source-only	0.545 $\pm$ 0.006	0.639 $\pm$ 0.018	0.651 $\pm$ 0.005	0.633 $\pm$ 0.006	0.646 $\pm$ 0.004
Mean matching	0.530 $\pm$ 0.005	0.628 $\pm$ 0.018	0.648 $\pm$ 0.006	0.634 $\pm$ 0.005	0.647 $\pm$ 0.006
CORAL	0.540 $\pm$ 0.006	0.633 $\pm$ 0.011	0.646 $\pm$ 0.004	0.632 $\pm$ 0.007	0.646 $\pm$ 0.005
MMD-rbf	0.530 $\pm$ 0.005	0.630 $\pm$ 0.014	0.646 $\pm$ 0.006	0.631 $\pm$ 0.005	0.644 $\pm$ 0.004
<i>Paired Δmacro-F1 vs. source-only (95% CI half-width, $\times 10^{-3}$)</i>					
Mean matching	−0.0150 (1.2)	−0.0104 (6.0)	−0.0026 (2.0)	+0.0004 (2.1) *	+0.0006 (2.1) *
CORAL	−0.0042 (0.9)	−0.0057 (8.2) *	−0.0047 (2.0)	−0.0010 (2.6) *	−0.0005 (1.6) *
MMD-rbf	−0.0145 (1.6)	−0.0083 (8.6) *	−0.0042 (1.9)	−0.0017 (2.6) *	−0.0024 (2.1)
<i>Q-factor at threshold 0.5</i>					
Source-only	0.780 $\pm$ 0.033	1.146 $\pm$ 0.046	1.195 $\pm$ 0.039	1.147 $\pm$ 0.041	1.172 $\pm$ 0.024
Mean matching	0.756 $\pm$ 0.033	1.164 $\pm$ 0.055	1.216 $\pm$ 0.040	1.174 $\pm$ 0.035	1.202 $\pm$ 0.039
CORAL	0.743 $\pm$ 0.028	1.182 $\pm$ 0.050	1.226 $\pm$ 0.033	1.179 $\pm$ 0.041	1.215 $\pm$ 0.030

That the overlap discussed above is the cause and not merely a correlate of capacity can be established by varying it. Figure 1c reports the excess gain for the full target minus the held-out target as the training fraction is swept from 0.1 to 0.9. For logistic regression, this gives 0.0000, 0.0000, −0.0001, −0.0001, and +0.0004; since the carrier has nothing to memorise, it must give zero in every case. For the random forest, it rises monotonically: +0.005, +0.016, +0.025, +0.027, +0.028, saturating as the held-out remainder shrinks. LightGBM and ExtraTrees lie in between and are noisier, with the latter falling back at the largest fraction where only a tenth of the sample remains held out. Manipulating the putative cause moves the effect in the predicted direction in the carriers for which this is predicted, and does not do so in the control.

Therefore, the inflation is carrier-dependent, and is also ordered by the property that forms the topic of this paper. Comparing the gain for CORAL measured in both ways at the 70% training fraction, the ratio is 1.0 for logistic regression, 1.4 for the MLP, 1.9 for LightGBM, 3.2 for ExtraTrees and 5.7 for random forest. When measured on the full

perturbed sample, part of what looks like a carrier gradient in adaptability is in fact a gradient in memorisation.

Therefore, we report every target-side quantity on the held-out target only. This correction reduces the effect of this benchmark by a factor of about six while leaving its structure intact, as the next subsection shows. Importantly, this pitfall is invisible when following the usual reporting, as the full-target numbers are internally consistent and reproduce across seeds with narrow intervals, but are in fact wrong.

This failure is an instance of the train–test contamination that Kapoor and Narayanan [14] documented across seventeen scientific fields, with the particularity that here it is introduced by the construction of the benchmark rather than by the handling of a pre-existing dataset; in addition, its size depends on which model is being evaluated. This topic also belongs to a line of work showing that in, domain adaptation and domain generalisation, the experimental protocol can dominate the differences between the algorithms under study. Gulrajani and Lopez-Paz [15] found that no domain generalisation method reliably beat a well-tuned baseline once model selection was held fixed, and Musgrave et al. [16] found that the accuracy differences between unsupervised domain adaptation algorithms were smaller than the differences induced by the validation procedure. Our result here is the same lesson reached from a different direction, with the model’s own capacity to memorise as the lever. That high-capacity learners fit their training sample exactly while still generalising is itself well documented [17,18]; what matters here is that the benchmark then scores them on that sample.

Table 3. Recovered fraction R of the shift damage, with its 95% bootstrap interval (ratio of means, percentile bootstrap, $B = 10^4$ resamples of the 15 carrier–seed pairs; Section 3.2). These are the intervals behind the R rows of Table 2. The MLP intervals span zero and are more than unit width; thus, the carrier is marked undetermined even though its recovery is not particularly uncertain, as its damage is too small to divide by.

Carrier	Mean Matching	CORAL	MMD-rbf
LR	0.00 [0.00, 0.00]	0.58 [0.55, 0.61]	0.89 [0.84, 0.94]
MLP	0.40 [−0.48, 1.18]	1.17 [−0.20, 2.62]	0.67 [−0.67, 1.96]
LightGBM	0.76 [0.63, 0.90]	0.98 [0.85, 1.13]	0.90 [0.76, 1.04]
ExtraTrees	0.92 [0.71, 1.15]	0.84 [0.69, 0.99]	0.91 [0.73, 1.07]
RF	0.86 [0.66, 1.05]	0.98 [0.79, 1.16]	0.85 [0.63, 1.10]

4.2. Shift Damage and Its Recovery

On held-out events, the shift is mild; however, it is not equally mild for every carrier. The unadapted AUC loss, which we call the shift damage D , is 0.0044 [0.0038, 0.0050] for logistic regression and 0.0034 [0.0025, 0.0042] for the MLP, against 0.0076 [0.0062, 0.0090] for LightGBM, 0.0074 [0.0062, 0.0086] for ExtraTrees and 0.0062 [0.0047, 0.0077] for random forest. A class-symmetric shift of the feature space costs a threshold-based carrier 1.4 to 1.7 times what it costs a hyperplane and 1.8 to 2.2 times what it costs the MLP. This is the first half of the carrier effect, which represents a statement about damage, not adaptation.

The second half is how much of the damage is then given back by feature alignment (Figure 2, Table 2). CORAL recovers 0.58 [0.55, 0.61] of the damage on logistic regression, 0.98 [0.85, 1.13] on LightGBM, 0.84 [0.69, 0.99] on ExtraTrees and 0.98 [0.79, 1.16] on random forest. On the MLP, the damage is too small for the ratio to be determined, and its bootstrap interval spans zero (intervals for all three methods are in Table 3). On the tree-based carriers, alignment returns essentially all of what the shift took away, while on the linear carrier it recovers a little over half.

The two halves also compound; in absolute terms, the gain for CORAL rises from +0.0025 [+0.0022, +0.0029] on logistic regression to +0.0075 [+0.0060, +0.0091] on Light-

GBM, $+0.0062$ [$+0.0046$, $+0.0078$] on ExtraTrees and $+0.0061$ [$+0.0040$, $+0.0082$] on random forest, for a factor of about 2.5 between the linear carrier and the tree ensembles. The ordering (linear $<$ neural $<$ trees) is preserved; however, the ordering within the tree family is not, with the three ensembles being mutually indistinguishable. This is not the case when scoring on the full perturbed sample, where random forest appears to profit most. This ordering is produced by the protocol of Section 4.1 rather than by the carriers themselves.

Mean matching provides a clean control at the bottom of the gradient. On logistic regression, its effect on the AUC is exactly zero to four decimal places on every seed, as the geometry requires. Translating the standardised features shifts every logit by a constant and leaves the ranking untouched. The real experiment reproduces an exact theoretical zero, which provides the strongest available check on whether the pipeline does what it is described as doing. On the tree carriers, mean matching recovers 0.76 to 0.92 of the damage, which is most of what CORAL recovers.

It is important to mention, first, that the absolute stakes at this shift intensity are small (less than 0.008 AUC even on the most affected carrier), so the practical question is not whether to adapt but whether the adaptation step is cheap, which is the case for the closed-form methods used here. Section 4.4 shows that the stakes grow with the intensity of the shift, reaching 0.026 AUC of damage at twice the headline intensity. A second qualification is that the gains in AUC do not transfer to every metric; at the default operating point, the point estimate of the macro-F1 change under CORAL is negative on all five carriers, significantly so on logistic regression (-0.0042) and LightGBM (-0.0047), and the Q-factor falls on logistic regression. Section 4.6 traces this discrepancy to the operating point rather than the feature space.

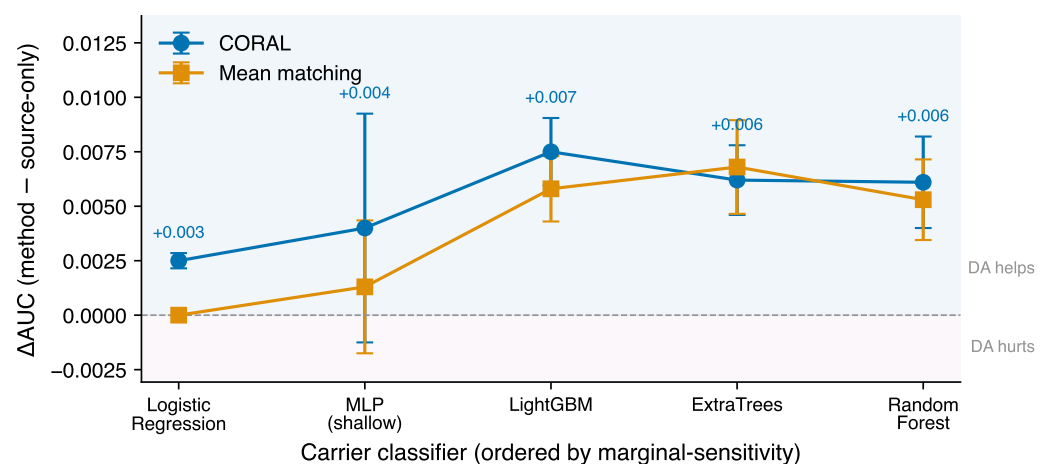


Figure 2. Carrier effect on the held-out target (15 seeds per cell; error bars are paired 95% confidence intervals). Both CORAL and mean matching yield larger gains on threshold-based carriers than on the hyperplane carrier, with the three tree ensembles mutually indistinguishable. Annotations give the CORAL Δ AUC. The dashed horizontal line marks Δ AUC = 0, that is, no change relative to the unadapted carrier; the band above it, where adaptation helps, and the band below it, where adaptation hurts, are shaded for orientation only.

4.3. Location Alignment Versus Covariance Alignment

Mean matching and CORAL differ only by the covariance term, so their difference isolates the value of second-order alignment. Table 4 confirms that the two methods do what they are supposed to do: mean matching leaves the Frobenius distance between the aligned-source and target covariance matrices exactly where source-only leaves it (0.258 ± 0.002), while CORAL reduces it to zero by construction, and the same holds for the correlation matrices (0.076 versus 0).

On the held-out target, this difference buys very little (Figure 3). The paired CORAL-minus-mean matching gap is $+0.0017$ [$+0.0004$, $+0.0031$] on LightGBM, -0.0006 [-0.0025 , $+0.0012$] on ExtraTrees and $+0.0008$ [-0.0006 , $+0.0022$] on random forest; one carrier favours CORAL by less than two parts in a thousand of AUC, one is compatible with zero, and one slightly favours mean matching. When scored on the full perturbed sample, the gap instead grows to $+0.009$ on the deepest carriers, which invites the explanation that deep trees exploit restored feature correlations. However, this is not the reason for this gap: covariance alignment in fact re-colours the source distribution more aggressively than pure translation, which moves memorised training events further back towards their memorised positions; thus, as the trees become deeper, there are more moving events.

Under a location-dominated class-symmetric shift, essentially all of the recoverable damage is recovered by matching the means; the covariance term is not harmful, but is not worth its extra assumptions either. The shift itself perturbs the correlation structure only mildly, with the largest single change being the LENGTH-CONC correlation from -0.76 to -0.74 , which is consistent with there being little for the covariance term to repair.

Table 4. Second-order statistics of the aligned source relative to the target in the standardised ten-feature space (mean $\pm$ std over 15 seeds). Frobenius norms for the differences of the covariance and correlation matrices, and Euclidean distance between the mean vectors.

	$\ \mathbf{C}_s - \mathbf{C}_t\ _F$	$\ \mathbf{R}_s - \mathbf{R}_t\ _F$	$\ \hat{\mu}_s - \mu_t\ _2$
Source-only	0.258 ± 0.002	0.076 ± 0.005	0.191
Mean matching	0.258 ± 0.002	0.076 ± 0.005	0.000
CORAL	0.000	0.000	0.000

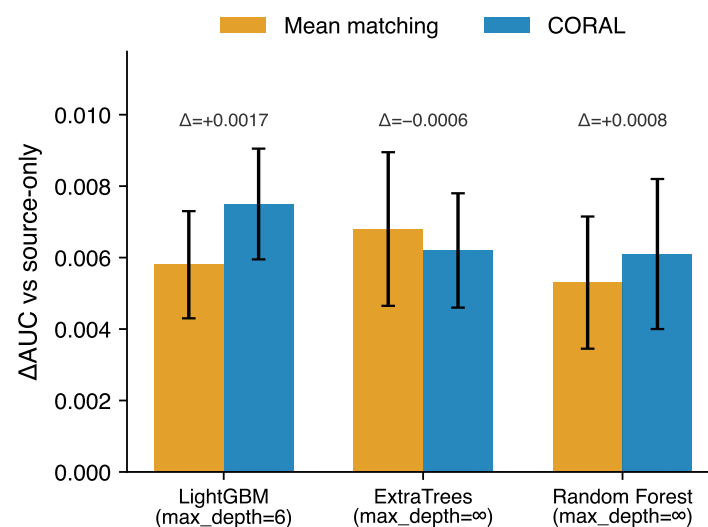


Figure 3. CORAL versus mean matching on the tree-based carriers (held-out target). The two methods are within two parts in a thousand of AUC of each other on all three ensembles. The dashed line marks $\Delta\text{AUC} = 0$; the annotation above each pair is the paired difference CORAL minus mean matching.

4.4. Ablations: Tree Depth, Shift Type, and Shift Intensity

Tree depth sweep. We sweep the maximum depth of the Random Forest carrier over $\{2, 4, 6, 8, 12, 16, \infty\}$ with the headline shift for src-only, mean matching, and CORAL, with fifteen seeds per cell (Figure 4). The gain for CORAL rises from $+0.0030$ at depth 2 to $+0.0063$ at depth 6, then is flat ($+0.0052$, $+0.0052$, $+0.0055$, and $+0.0061$ at depths 8, 12, 16, and unbounded, respectively). The source-only AUC saturates over the same range (0.692 at depth 2, 0.759 unbounded). Thus, depth matters while the trees are still too shallow

to place enough thresholds to be displaced by the shift, and stops mattering once they are deep enough; there is no monotone growth, and the gap between CORAL and mean matching stays within ± 0.0008 at every depth. When scored on the full perturbed sample, the same sweep shows tenfold growth with depth and the advantage of CORAL opens up beyond depth 8. Both grow with depth, for the same reason that memorisation does, and neither is present on held-out events.

LightGBM without a depth cap. The configurations used here (Appendix A) cap LightGBM at depth 6 while leaving random forest and ExtraTrees unbounded. This cap is a deliberate choice; since LightGBM grows trees without a depth limit unless one is set, it is worth evaluating with no limit. With $\text{max_depth} = -1$ and $\text{num_leaves} = 255$ in place of the default 31, the control changes two settings and not depth alone; the source-only held-out AUC then falls slightly, from 0.765 to 0.759, and the gain for CORAL is unchanged within its interval, $+0.0069$ [$+0.0052, +0.0087$] against $+0.0075$ [$+0.0060, +0.0091$]. This shows that the position of LightGBM in the carrier ordering is not an artefact of the cap, as it is consistent with the plateau in the depth sweep.

Shift type and intensity. We re-run all five carriers with src-only, mean matching, and CORAL under four alternative shifts, with five seeds per cell (Figure 5): (i) the atmospheric attenuation family alone (a multiplicative rescaling, which the $\log_{10}$ transform turns into a pure translation for SIZE but not for WIDTH and LENGTH, so the family is location-dominated rather than purely translational; see Appendix C and the pure translation ablation below); (ii) the NSB-injection family alone (additive noise, which inflates the spread of the two concentration features, $+0.027$ and $+0.026$ of standardised standard deviation for CONC and CONC1, respectively, while moving the means by little; the norm of the mean shift is 0.018 against 0.191 for the headline composite, and the change in the spread of WIDTH and LENGTH is below 0.001, see Appendix C); and the headline composite at (iii) half intensity and (iv) double intensity. Four observations follow:

- (i) The location-only shift reproduces the pattern of the headline composite, as it should, since the headline composite is location-dominated: damage 0.0036 on logistic regression and 0.0073–0.0080 on the trees, with CORAL recovering $+0.0025$ and $+0.0070$ – $+0.0085$, respectively.
- (ii) The noise-only shift barely damages the carriers at all on held-out events (0.0009 to 0.0020 of AUC and -0.0003 on ExtraTrees), and alignment recovers nothing measurable ($|\Delta\text{AUC}| \leq 0.0012$ on every carrier). Additive noise degrades the information content of the features rather than displacing them, so there is nothing for a marginal-alignment method to restore. This represents the “washes out” cell of the framework. The held-out evaluation strengthens this result, as not only does alignment fail to help, there is almost nothing to help with.
- (iii) A genuinely pure translation separates the two components of the location-only family. We build a target by adding a constant per-feature offset to the standardised source. The offset is equal to the mean shift of the headline composite, so the covariance is unchanged by construction. We then repeat the protocol with five seeds. Logistic regression has damage and gain of exactly zero, to five decimal places and on every seed, for both mean matching and CORAL; a translation of the feature space is invisible to a linear score’s ranking, so there is nothing to lose and nothing to repair. The tree carriers are damaged (0.0078 [0.0051, 0.0104] on LightGBM, 0.0056 [0.0020, 0.0091] on random forest), but recover essentially all of it ($+0.0074$ and $+0.0051$ with CORAL). The contrast with the location-only family, where logistic regression still loses 0.0036 of AUC, measures how far that family is from a pure translation; its multiplicative rescaling is translational for the log-transformed SIZE, but not for WIDTH or LENGTH (Appendix C).

- (iv) The stakes scale with intensity, and faster than linearly. Halving the headline intensity leaves damage of 0.0018–0.0032; doubling it raises damage to 0.0125 on logistic regression, 0.0141 on the MLP, and 0.0263 on LightGBM, with CORAL recovering +0.0080, +0.0107, and +0.0199. At double intensity, the effect that is worth a few parts in a thousand at the headline setting is worth two AUC points on a tree carrier, while the carrier ordering is unchanged. Thus, the mildness of the headline result is a property of the chosen perturbation strength, not of the mechanism.

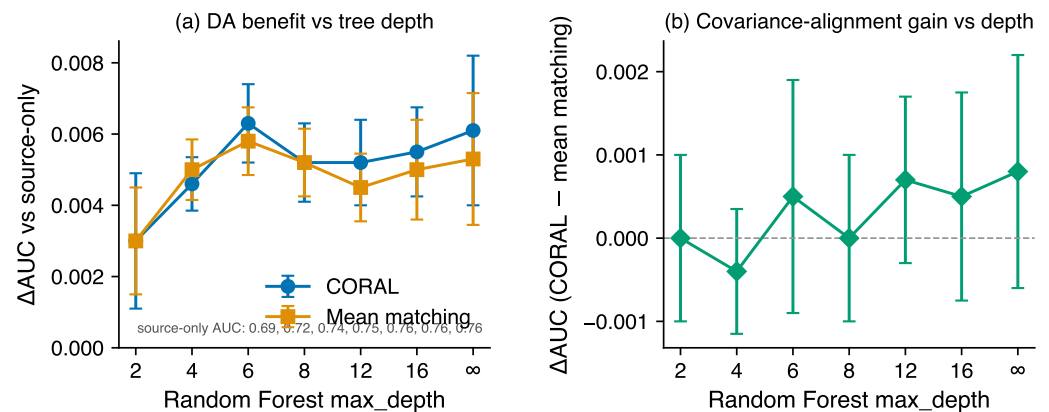


Figure 4. Random forest depth sweep on the held-out target (15 seeds per cell, paired 95% CIs): (a) Δ AUC of CORAL and mean matching versus maximum tree depth, with the src-only AUC at each depth listed in the panel; (b) paired difference CORAL minus mean matching, which is within ± 0.0008 at every depth. The dashed horizontal line in each panel marks zero, that is, no difference between the two methods being compared in that panel.

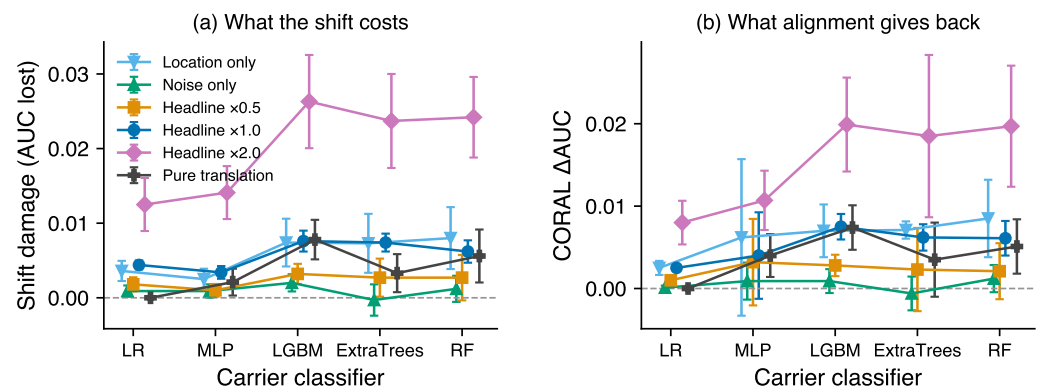


Figure 5. CORAL Δ AUC on the held-out target under different shift types and intensities (5 seeds per cell except the headline $\times 1.0$ reference, 15 seeds; paired 95% CIs). The carrier pattern is preserved and its amplitude grows with intensity; the noise-only shift produces no benefit on any carrier. The dashed horizontal line in each panel marks Δ AUC = 0.

4.5. Marginal Sensitivity Measured Without Domain Adaptation

The ordering of carriers used so far rests on a structural argument. To avoid arguing in a circle, Section 3.5 defines an index $S(\delta)$ that involves neither alignment nor target data, consisting of the AUC lost by the source-only model on held-out source events when every feature is translated by $\pm\delta$. Figure 6a shows the result. Logistic regression sits at exactly zero for every δ , as it must; the MLP is small ($S(0.25) = 0.012 \pm 0.004$); and the three tree ensembles are clustered well above it (ExtraTrees 0.047 ± 0.004 , LightGBM 0.060 ± 0.004 , and random forest 0.060 ± 0.005), losing roughly a tenth of their AUC at a translation of a quarter of a standard deviation. From a measurement that says nothing about domain adaptation, the index reproduces the same partition that the framework asserts.

The index also predicts the two quantities from Section 4.2: across the five carriers, $S(0.25)$ is correlated with the shift damage at Pearson $r = 0.86$ and with the CORAL gain at $r = 0.96$ (Spearman $\rho = 0.60$ and 0.70 , respectively, over the same five points). Pooling the 75 individual carrier–seed pairs, the correlation with the gain falls to $r = 0.30$, since within a carrier the seed-to-seed variation of the index is not informative about that seed’s gain. Both figures describe this benchmark; Section 5.4 clarifies the scope.

The flip rate at threshold 0.5 tells a complementary threshold-dependent story: under the same translation, logistic regression flips 9% of its hard predictions at $\delta = 0.25$ and 39% at $\delta = 1$, more than any tree at large δ , while its ranking is unchanged. A linear carrier is marginal-robust in its ranking and marginal-fragile in its operating point, which is the distinction that Section 4.6 makes quantitative.

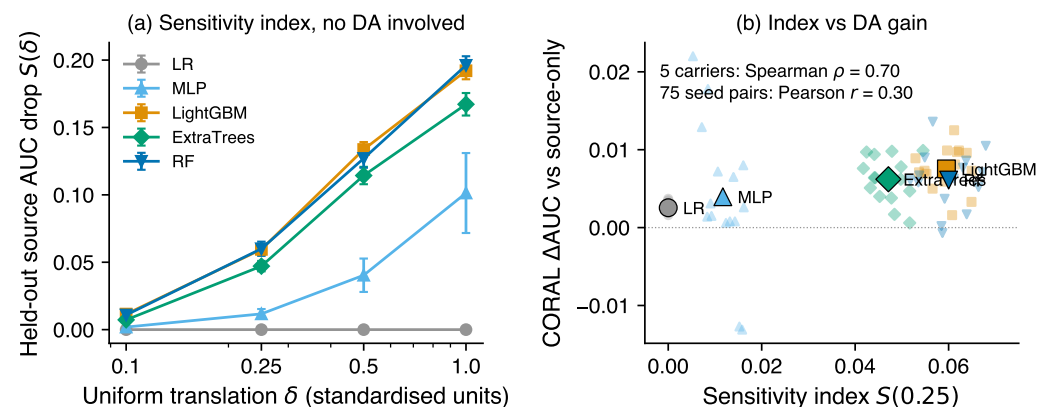


Figure 6. Marginal sensitivity measured without domain adaptation. (a) AUC lost on held-out source events by the source-only model when every standardised feature is translated by $\pm\delta$ (mean over the two directions, $\pm$ std over 15 seeds); logistic regression is identically zero. (b) CORAL Δ AUC against the index at $\delta = 0.25$; small markers are individual seeds, large markers are the carrier means. The dotted horizontal line in (b) marks Δ AUC = 0.

4.6. Q-Factor Across Operating Points

The Q-factors quoted so far are evaluated at the default threshold of 0.5. Because the score distribution of a carrier changes under shift and under alignment, a fixed threshold can exaggerate or hide a gain. Following the protocol of Section 3.6, Figure 7 shows $Q(\theta)$ on the held-out target over the whole threshold range. Table 5 reports the paired gain at three operating points: the default, the threshold θ^* that maximises Q on the source validation split (which is the choice available to a practitioner without target labels), and the oracle maximum on the target.

Logistic regression is the clearest case, and resolves the apparent contradiction in Table 2. At the default threshold, CORAL lowers its Q-factor by 0.037 $[-0.044, -0.031]$, which taken alone would indicate that alignment harms a linear carrier. At $\theta^* = 0.20$, which both the unadapted and adapted models select from source data alone, the ordering reverses and CORAL is worth $+0.009 [+0.007, +0.011]$, a gain of under one per cent but with the sign restored. Alignment moves the intercept of a linear score; the default threshold no longer sits where it did, and re-selecting it on source data removes the damage entirely. For a marginal-robust carrier, the shift displaces the operating point, not the ranking, so the operating point is what should be re-selected.

On the tree carriers, the gain at the source-selected threshold is larger than at the default, not smaller: $+0.104 [+0.073, +0.137]$ on LightGBM and $+0.076 [+0.028, +0.123]$ on random forest, against $+0.030$ and $+0.043$ at $\theta = 0.5$. In relative terms, CORAL improves the Q-factor of LightGBM by 8.5% and that of random forest by 6.1% at the operating

point that a practitioner would actually choose. Mean matching is again close behind, and slightly ahead on LightGBM.

The observation-time translation of Section 5.5 follows from these numbers rather than from the default threshold ones. A 6% improvement in Q at fixed exposure corresponds to reaching a given significance $1/1.06^2 \approx 89\%$ of the time, that is, a saving of about one hour in ten at the headline shift intensity. When computed at the default threshold on the full perturbed sample, the same quantity would suggest savings of more than half the observation time, which is an artefact of both choices.

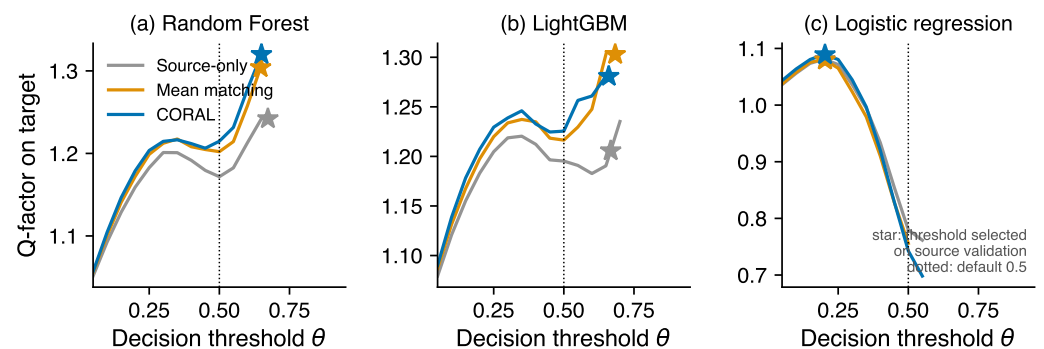


Figure 7. Q-factor on the held-out target as a function of the decision threshold (mean over 15 seeds) for source-only, mean matching, and CORAL. Stars mark the threshold that maximises Q on the source validation split; the dotted line is the default 0.5. Curves stop where fewer than 50 hadrons are accepted.

Table 5. Paired CORAL gain in Q-factor relative to source-only on the held-out target at three operating points (mean over 15 seeds, 95% paired intervals). Each model uses its own source-selected threshold: θ^* maximises Q on the 30% source validation split for that model and is then transferred unchanged to the target, so the two leftmost numeric columns need not coincide. $Q_{\text{src-only}}$ is quoted at the source-only θ^* . Oracle refers to the best threshold on the held-out target itself, which no practitioner can select.

Carrier	θ^* (Source-Selected)		$Q_{\text{src-only}}$ at θ^*	Paired ΔQ , CORAL Minus Source-Only		
	src-only	CORAL		at 0.5	at θ^*	Oracle
Logistic regression	0.20	0.20	1.081	−0.037 [−0.044, −0.031]	+0.009 [+0.007, +0.011]	+0.009 [+0.007, +0.011]
MLP-shallow	0.44	0.54	1.229	+0.037 [+0.016, +0.057]	+0.031 [−0.004, +0.066]	+0.029 [−0.004, +0.062]
LightGBM	0.67	0.66	1.220	+0.030 [+0.012, +0.048]	+0.104 [+0.073, +0.137]	+0.086 [+0.057, +0.116]
ExtraTrees	0.52	0.53	1.214	+0.033 [+0.016, +0.049]	+0.029 [+0.005, +0.052]	+0.021 [+0.001, +0.041]
Random Forest	0.67	0.65	1.254	+0.043 [+0.031, +0.055]	+0.076 [+0.028, +0.123]	+0.084 [+0.047, +0.121]

4.7. Measured Shift Diagnostics

The S0 headline composite produces a statistically detectable but mild shift in the preprocessed feature space (Table 6). The kernel MMD between source and target is 0.0011 ($p = 0.005$, permutation test with $B = 1000$ permutations, of which four exceeded the observed value), indicating a small but significant distributional difference. The proxy A-distance is 0.24 (domain classifier accuracy 56%), confirming modest distinguishability. These diagnostics are computed on the feature distributions and so are unaffected by the evaluation protocol question in Section 4.1; they describe the perturbation, and their smallness is consistent with the small damage measured on held-out events.

The calibration drift is negligible ($\Delta\text{ECE} = 0.002$), indicating that the logistic regression classifier's probability estimates remain well calibrated on the shifted target. This is consistent with the class-symmetric nature of the S0 shift: the perturbation transports both classes identically, so the class-conditional structure is not altered relative to the decision rule even though the features themselves have moved.

Table 6. Scalar shift diagnostics for the S0 headline composite, computed on the preprocessed ten-feature space.

Diagnostic	Value	Note
Kernel MMD (RBF, median heuristic)	0.0011	$p = 0.005$, $B = 1000$ permutations
Proxy A-distance	0.24	domain-classifier accuracy 56%
Calibration drift (ΔECE , logistic regression)	0.002	$\text{ECE } 0.017 \rightarrow 0.019$

4.8. Robustness and Sanity Checks

Each cell of the benchmark was run with fifteen independent random seeds controlling the train/test split and classifier initialisation. Across the twenty cells of Table 2, the standard deviation of the held-out AUC across seeds runs from 0.0045 to 0.0085; the paired intervals on the differences are considerably narrower because the seed is shared between the compared cells.

MMD alignment with an RBF kernel is the one method in the benchmark that can in principle match statistics beyond second order; it lands within 0.002 AUC of CORAL on every carrier (Table 2). On the tree and neural carriers, the paired difference ranges from -0.0017 to $+0.0005$; every interval contains zero. On logistic regression this is not the case, the difference being $+0.0014$ [$+0.0011$, $+0.0016$] in favour of the kernel method. This difference is not explained by closer second-moment matching; CORAL brings the standardised spread of the aligned source onto the target's to within 0.0001 on WIDTH, LENGTH and SIZE, whereas MMD-rbf leaves residuals of 0.013, 0.004, and 0.004 on the same three features, so the method that matches the second moment worse is the one that scores better, and its transform departs less from the identity ($\|\mathbf{W} - \mathbf{I}\|_F = 0.080$ against 0.146). Therefore, we record the gap as measured but not explained.

The complete results, for every carrier and every method, are those of Table 2.

5. Discussion

5.1. A Two-Axis Framework for DA Effectiveness

The results in Section 4 motivate a two-axis framework for reasoning about when feature alignment domain adaptation helps, hurts, or washes out (Figure 8).

The first axis is the **nature of the shift**. The S0 benchmark implements a class-symmetric shift in which both classes are transported by the same map; thus, the marginal $P(\mathbf{x})$ moves, while the class-conditional structure is carried along with it rather than being deformed. Feature alignment methods, which target $P(\mathbf{x})$, can only recalibrate marginal statistics, but cannot improve the decision boundary itself; in contrast, class-asymmetric shifts, where $P(\mathbf{x} | y=1)$ and $P(\mathbf{x} | y=0)$ shift differently, would displace the optimal decision boundary, giving adaptation methods a genuine signal to recover. In such a regime, the sign of the effect does not follow from this argument alone; aligning the marginals can also erode the feature–label dependence that a class-asymmetric shift leaves intact, so the signal being gained and the structure being lost pull in opposite directions. Nothing that we measure here settles the question of which of these is dominant [19].

The second axis is the **carrier's marginal sensitivity**. Our results demonstrate that even under a class-symmetric shift that leaves the classes in the same relation to one another,

the choice of carrier determines how much the shift costs and how much of that cost is returned by alignment. This axis is the primary empirical contribution of this paper.

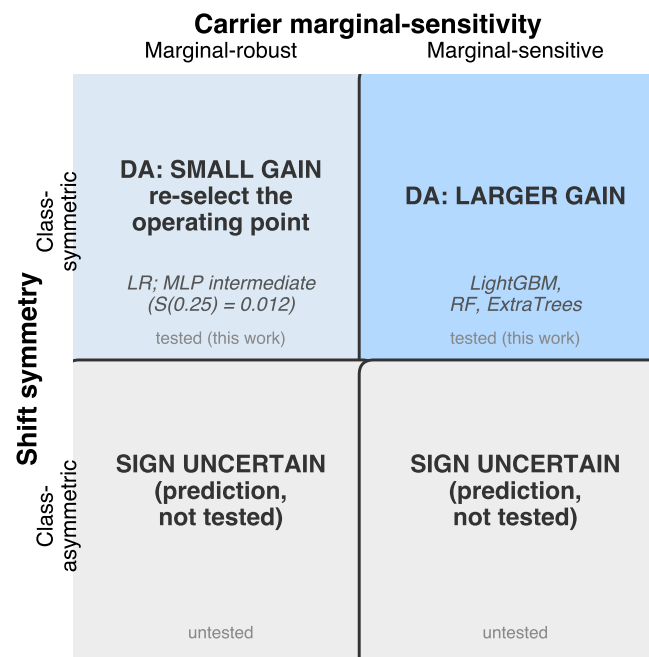


Figure 8. Conceptual framework. The effectiveness of domain adaptation depends jointly on the symmetry of the source–target shift and the marginal sensitivity of the carrier classifier. Only the top row (class-symmetric shift) is supported by the experiments of this paper; the bottom row states untested predictions that follow from the argument of Section 5.1 and remain to be validated on class-asymmetric shifts.

5.2. Why Trees Are Marginal-Sensitive

Figure 9 illustrates the mechanism underlying the marginal-sensitivity axis. A linear classifier (e.g., logistic regression) partitions the feature space with a hyperplane. The orientation of this hyperplane depends on the relative geometry of the two classes. A uniform marginal shift translates both classes equally, which leaves the orientation of the hyperplane and the induced ranking of events untouched while displacing the hyperplane relative to the data. Thus, a decision taken at a fixed score threshold moves. The classifier is marginal-robust in its ranking, meaning that its decision geometry is invariant to location shifts.

The marginal sensitivity of tree ensembles is consistent with their known brittleness to small absolute perturbations of individual features. Kantchelian et al. [20] showed that tree ensembles can be evaded by adversarial inputs that displace a few features across learned thresholds, which is precisely the mechanism that a uniform marginal shift triggers on every event at once. A tree-based classifier partitions the feature space with axis-aligned threshold splits. Each split is a function of the absolute value of a single feature: if $x_j < t$, go left; else, go right. A uniform marginal shift displaces the feature distribution relative to the fixed threshold t , causing events that were correctly classified on the source to be misclassified on the target. The classifier is marginal-sensitive, that is, its decision boundaries are functions of the marginal feature distributions.

Feature alignment DA methods (CORAL, mean matching, MMD) recalibrate the marginal distributions, effectively shifting the target features back to the source’s coordinate system. For tree-based carriers, this restores the alignment between the feature distributions and the learned thresholds. For linear carriers, the translational part of the shift is invisible to the ranking, so only the non-translational part is left for alignment to repair. This is

why the gain is smaller but not zero; the losses that appear at a fixed operating point are a displacement of the threshold rather than of the geometry, and are removed by re-selecting it on source data (Section 4.6).

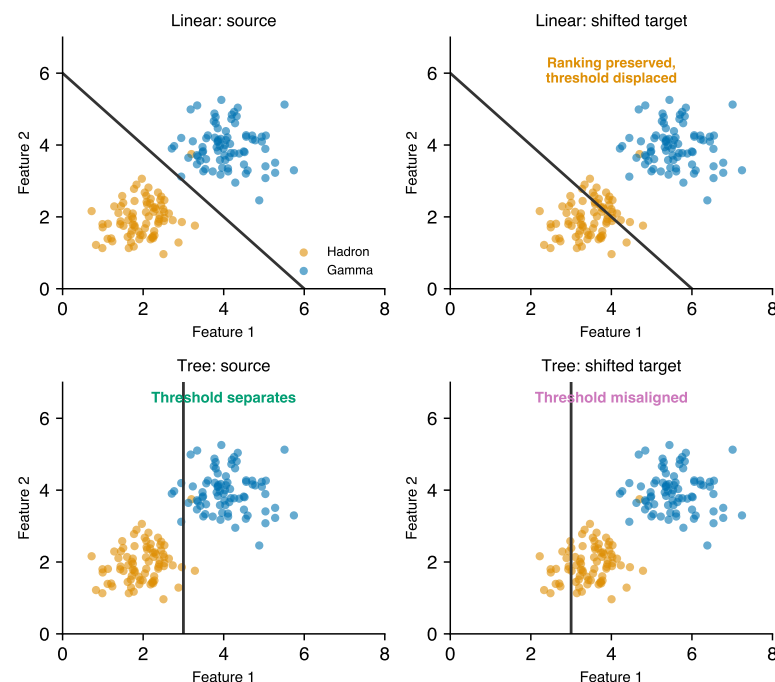


Figure 9. Schematic of the marginal-sensitivity mechanism. (**Top row**): After a uniform marginal shift, the fixed linear boundary is crossed but the ranking induced by a linear classifier is unchanged. (**Bottom row**): A tree classifier’s axis-aligned splits become misaligned with the feature distribution after the same shift, causing misclassification. Feature alignment restores that agreement in the bottom row. In the top row, the shift displaces the score threshold rather than the geometry, and the loss is recovered by re-selecting the threshold on source data rather than by alignment.

5.3. Effects of Tree Depth

A natural reading of the carrier ordering is that depth amplifies marginal sensitivity without limit: each split sits at an absolute feature location, so a longer cascade of splits should compound the location dependence, and a forest with unbounded depth should profit from alignment more than a depth-capped booster. A depth sweep scored on the full perturbed sample supports exactly this reading, with the benefit growing tenfold from depth 2 to unbounded depth.

Held-out evaluation does not support the same conclusion; the benefit rises while the forest is shallow (roughly doubling between depth 2 and depth 6), then is flat out to unbounded depth (Section 4.4, Figure 4). Removing the depth cap from LightGBM likewise leaves its benefit unchanged. The monotone growth seen on the full perturbed sample is an artefact of the evaluation protocol of Section 4.1, and grows with depth for the same reason that memorisation does.

Marginal sensitivity requires two things: that the decision rule reference absolute feature locations, and that it place enough of them to be displaced. Only the first requirement distinguishes the families; a forest with depth 2 satisfies the first but not the second, and accordingly behaves like the linear carrier (+0.0030 [+0.0011, +0.0049] against +0.0025 [+0.0022, +0.0029]). Beyond modest depth, the second condition is met and further splits add nothing. The same evidence disposes of the similar idea that covariance alignment pays off on deep trees: mean matching and CORAL are within two parts in a thousand of AUC at every depth (Figure 4b), so whatever deep trees do with feature correlations, restoring these correlations is not what the alignment step is providing.

5.4. Limitations

Synthetic shift only. The S0 benchmark uses a class-symmetric synthetic perturbation. Real-world distribution shifts are typically class-asymmetric (e.g., instrumental effects that affect signal and background populations differently). The bottom row of the 2×2 framework (Figure 8) remains a prediction to be validated on real-data scenarios. In that regime, methods that align the joint feature–label distribution rather than the marginals alone, such as the joint-MMD formulation of Long et al. [21], are the natural candidates, since blind marginal alignment can erode the feature–label dependence that a class-asymmetric shift leaves intact.

Same base events in both domains. Source and target are built from the same base events by deterministic perturbation. This is a deliberate design choice, as it guarantees that the only difference between the domains is the perturbation itself; in this way, any change in target performance is attributable to the shift and the carrier’s response to it, with no confounding from event-level reweighting, class balance drift, or sampling noise between two independent samples. The price is that the two domains are not statistically independent, from which two consequences follow: first, the shift diagnostics in Section 4.7 measure only the perturbation, which is why they are small; second, the setting is more favourable to feature alignment than a real sim-to-real gap, in which the target would also differ in event population; therefore, the gains reported here are an upper bound on what the same perturbation would yield across independent samples, and the ordering of carriers is the transferable result rather than the absolute size of the gains. The target labels are never used, and the carrier is trained only on the 70% source split and evaluated on the perturbed copies of the 30% held-out events, so neither target labels nor training events reach the reported figures.

What the confidence intervals mean. The fifteen seeds vary the train/test split and classifier initialisation on one fixed base sample. As such, the paired intervals quantify the reproducibility of each ΔAUC under resampling of the split and the training randomness. This indicates that the ordering of carriers is not an artefact of a particular split, but does not mean that it would survive a change of dataset, instrument, or perturbation family. Statements of significance in this paper are intended to be read within this scope. The very small intervals on the tree carriers reflect the near-determinism of the pipeline once the base sample is fixed, and should not be mistaken for precision about the population of possible datasets.

Five carriers but only one association. The correlation between the sensitivity index and the CORAL gain (Section 4.5) is computed over five classifier families and is a description of the ordering observed in this benchmark, not an estimate of a population parameter or a causal claim. What supports the mechanism is the convergence of independent lines of evidence, the DA-free sensitivity index, the depth sweep within a single family, the shift type ablation, and the covariance analysis rather than the value of any single correlation coefficient. Extending the set of carriers (kernel SVMs, k -nearest neighbours, single trees, gradient boosting with varied regularisation) is a natural next test of the ordering.

Carrier hyperparameters. The main matrix compares each carrier in the configuration listed in Appendix A, which is a conventional setting rather than a tuned one. The configurations are not library defaults, and differ from one another in a relevant way: random forest and ExtraTrees grow trees to unbounded depth, which is the scikit-learn default, whereas LightGBM is capped at depth 6, which is a deliberate choice since LightGBM is unbounded by default. The gradient of Section 4.2 mixes two effects, namely, the ensemble family and the depth. Two controls separate them: the depth sweep (Section 4.4) holds the family fixed and shows that depth moves a random forest from the linear carrier’s position to the tree carriers’ position, saturating once the forest is deep enough to place a sufficient

number of thresholds; conversely, LightGBM with the depth cap removed ($\text{max_depth} = -1$, 255 leaves; Section 4.4) tests whether the boosted family moves up the gradient when the cap is removed. Neither control changes the conclusion, but the reader should take the main matrix ordering as referring to “carriers as commonly configured” rather than as a statement about ensemble families per se.

Feature alignment methods only. We evaluate CORAL, mean matching, and MMD, all of which are marginal-alignment methods. Adversarial methods (DANN [22]), self-training, and test-time adaptation may interact differently with the carrier sensitivity axis.

Tabular features only. The benchmark operates on tabular Hillas parameter features. Image-level domain adaptation (e.g., on convolutional features) may exhibit different carrier sensitivity patterns, particularly for architectures with batch normalisation.

Location-dominated shifts. The headline composite is location-dominated, and the shift type ablation (Section 4.4) shows that the carrier gradient is a property of location shifts. A pure noise shift yields no benefit on any carrier. Shifts that alter the class-conditional structure or that combine location and noise components in different proportions than those tested here might place carriers differently along the gradient. The ablation covers two perturbation families, a pure translation, and two intensities; the remaining families (flat-fielding, zenith, energy threshold) were not swept.

5.5. Connection to Applied Domains

What transfers to real CTA observations and what does not. The perturbation families of the benchmark are modelled on the three effects that dominate the simulation-to-observation gap in IACT analysis, namely, night sky background, atmospheric transmission, and flat-fielding, and act on the Hillas parameters in the same way that those effects act on real images (Appendix B). However, real observations differ from simulations in ways that the benchmark does not reproduce: the gamma and proton populations are sampled from different spectra and different fields of view, the background contains electrons and muons that the simulation omits, the instrument response drifts within a night, and most importantly, these effects are not identical for the two classes, so the shift is not class-symmetric. Two of our conclusions are expected to carry over, while one is not. The ordering of carriers by marginal sensitivity is a property of the classifiers, not of the data, and the depth sweep shows that it holds within a single family; thus, we expect it to survive on real data. The finding that a pure noise shift is not repaired by alignment is likewise structural. We do not expect the absolute size of the gains or the sign of the effect on a linear carrier to transfer, as both of these depend on the shift being a translation that leaves the class-conditional structure intact. For a CTA analyst, tree-based gamma/hadron separators trained via Monte Carlo are the classifiers most likely to profit from a label-free alignment step, and the size of the benefit must be measured on the instrument’s own data.

The benchmark is instantiated on IACT gamma ray astronomy data, where the source–target gap between Monte Carlo simulations and real observations is a recognised operational challenge [1,23]. In this domain, tree-based classifiers (random forest, BDT, LightGBM) are the prevalent choice for gamma/hadron separation, and MC-trained models are routinely deployed on real data without domain adaptation. Our results indicate that this practice leaves recoverable classification performance on the table.

The headline effect size is operationally meaningful under the stated assumptions. In the background-dominated regime that applies to faint sources, the Li & Ma significance [24] reduces to $S \simeq N_{\text{ex}} / \sqrt{\alpha(1 + \alpha) N_{\text{off}}}$, where $N_{\text{ex}} = \epsilon_{\gamma} N_{\gamma}$ is the excess after cuts, $N_{\text{off}} \propto \epsilon_h N_h$ is the background after cuts, and α is the ON/OFF exposure ratio. Both counts scale linearly with the observation time t , so $S \propto (\epsilon_{\gamma} / \sqrt{\epsilon_h}) \sqrt{t} = Q\sqrt{t}$. At fixed exposure, the significance is proportional to the Q-factor, and the time to reach a given significance

scales as Q^{-2} . We take the operating point at the threshold that a practitioner could actually choose, which is the point that maximises Q on source validation data (Section 4.6). At this point, CORAL raises the Q -factor of random forest from 1.254 to 1.330, for a relative gain of 6.1%; the same relative gain for LightGBM is 8.5%. A 6% improvement in Q at fixed exposure is a 6% improvement in significance, or equivalently, the same significance in $1/1.06^2 \approx 89\%$ of the observation time, which for a 50-h campaign is about five hours. This is a real but modest saving, and is obtained for the cost of a closed-form transform fitted on unlabelled data. It also scales, though not proportionally: at twice the headline intensity, the damage on a tree carrier is 3.2 to 3.9 times larger, the recovered share falls rather than rises (from 0.98 to 0.76 on LightGBM), and the absolute AUC returned grows by a factor of 2.7 to 3.2. A Q -factor measured at the default threshold on a target containing the classifier's own training events would instead suggest savings of more than half the observation time; thus, both choices inflate it.

5.6. Practical Recommendations

Based on our findings, we offer four practical recommendations for practitioners using supervised classifiers under distribution shift:

1. **Evaluate DA with the deployment carrier.** The benefit of domain adaptation depends on the carrier classifier, so a result obtained on one model family does not transfer to another. The sizes involved should be judged against the shift. At the headline intensity of this benchmark, the whole effect is worth less than 0.008 AUC even on the most affected carrier, while at twice that intensity it reaches about 0.02. Evaluating only on a linear carrier understates what the same alignment is worth on a tree ensemble by a factor of about two and a half, which does not change with intensity.
2. **When using trees, even simple mean matching helps.** The location alignment component accounts for the majority of the DA benefit on tree-based carriers. Mean matching is hyperparameter-free and computationally trivial.
3. **Covariance alignment is optional under a location-dominated class-symmetric shift.** The paired gap between CORAL and mean matching is within about 0.002 AUC on all three tree carriers, reaching significance only on LightGBM ($+0.0017$ [$+0.0004, +0.0031$]); it also stays within ± 0.0008 at every depth of the random forest sweep. Mean matching captures essentially all of the recoverable damage, so the covariance term is worth adding only if the shift is known to alter the correlation structure, which in this case it barely does (Table 4).
4. **On a linear carrier, expect a smaller gain and re-select the operating point.** A uniform translation leaves the ranking of a linear score unchanged, so only the non-translational part of the shift is available for alignment to repair: CORAL gains $+0.0025$ [$+0.0022, +0.0029$] of AUC on logistic regression and RBF-kernel MMD $+0.0039$ [$+0.0034, +0.0044$], against $+0.006$ to $+0.008$ on the tree carriers. The losses that appear at a fixed threshold are -0.004 of macro-F1 and -0.037 of Q -factor, representing a displacement of the operating point as opposed to the geometry. Re-selecting the threshold on source data turns the Q -factor loss into a gain of $+0.009$ [$+0.007, +0.011$] (Table 5).

6. Conclusions

We set out to test whether the benefit of feature alignment domain adaptation transfers across downstream classifiers. We find that it does not, though for reasons partly different from what we expected.

Under a controlled class-symmetric location shift evaluated on events that no carrier has seen, the carrier matters at two points: it first sets how much the shift costs (0.0034

to 0.0044 of AUC for the hyperplane and neural carriers, 0.0062 to 0.0076 for the three tree ensembles), then also sets how much CORAL gives back (58% of the loss on logistic regression against 84–98% on the trees). The two compound into an absolute benefit that is some 2.5 times larger on a tree carrier than on a linear one. The second half of the above statement is a property of both the method and the carrier: RBF-kernel MMD recovers 0.85 to 0.91 on every carrier for which the ratio is determined; thus, it is CORAL and mean matching, not feature alignment as such, for which the carrier matters twice. The mechanism is the structural difference between hyperplane classifiers, where the ranking is invariant to a uniform translation, and threshold-split classifiers, where the decision boundaries sit at absolute feature locations. A measurement of this sensitivity taken without any adaptation and without target data will reproduce the carrier ordering and be correlated with both the damage and the recovery.

Two effects that a full-target protocol would report are absent when the target is restricted to unseen events. Feature alignment does not become more valuable as trees grow deeper: the benefit saturates by depth six and is flat thereafter. Covariance alignment does not outperform simple mean matching on deep trees: the two differ by less than two parts in a thousand of AUC on every ensemble and at every depth. Both apparent effects arise from scoring the adapted model on a target that contains perturbed copies of its own training events, which rewards memorisation in proportion to capacity; capacity is exactly the variable that distinguishes the carriers.

That observation is the part of this work we expect to be most useful elsewhere. In any benchmark where the target domain is a perturbed copy of a sample that the model was partly trained on, the measured benefit of adaptation is inflated by a factor that tracks the model's capacity to memorise. This factor ranges from 1.0 for a linear model to 5.7 for an unbounded-depth random forest. Such a benchmark will report a carrier gradient regardless of whether or not one actually exists, and its numbers will be internally consistent and reproducible while still being wrong. The remedy is cheap: score only on target events that the model has never seen in any form.

The practical recommendations that follow are correspondingly modest. Evaluate domain adaptation with the carrier you intend to deploy, since the benefit is not transferable between model families. If that carrier is tree-based, apply at least mean matching, which is free and recovers most of what covariance alignment recovers; if it is linear, expect alignment to return only part of the ranking loss (a little over half of it under CORAL) and re-select the operating point, since in a linear model a class-symmetric shift mainly displaces the threshold rather than the geometry. Finally, judgment should use the applicable shift magnitude: at the headline intensity of this benchmark, the whole effect is worth less than 0.008 AUC, while at twice that intensity the damage on a tree carrier reaches 0.026 and the recoverable amount of it becomes operationally interesting.

Future work will extend the framework to class-asymmetric shifts, for which we deliberately do not predict a sign here. Such a shift displaces the optimal decision boundary, which gives alignment something to recover; however, marginal alignment can also erode the feature–label dependence that the shift leaves intact, and whether the first effect or the second dominates is not settled by anything we have measured so far [19]. Future work is also needed to extend the comparison to adversarial and self-training methods as well as to an actual sim-to-real pair once public data exposing a common feature space become available.

Funding: This research received no external funding.

Data Availability Statement: The code, Hydra configurations, synthetic perturbation framework, per-seed result manifests, and figure generation scripts used in this study are available at <https://doi.org/10.3390/a19100859>.

[//github.com/antopag/cherenkov-sim2real](https://github.com/antopag/cherenkov-sim2real) (accessed on 6 October 2026); the archived release of that repository, which is the state of the code that produced every number, table and figure reported here, is deposited on Zenodo under DOI 10.5281/zenodo.23257260 (<https://zenodo.org/records/23257260>, accessed on 6 October 2026). The CTA Prod5 public Monte Carlo event lists are archived by the CTA Observatory and Consortium on Zenodo [12] (<https://zenodo.org/records/7298569>, accessed on 6 October 2026).

Acknowledgments: This research made use of the CTA DL1 and DL2 event lists provided by the CTA Observatory and Consortium (version prod5-DL2-release1-DL2).

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

Appendix A. Carrier Classifier Hyperparameters

Table A1 reports the hyperparameters used for each carrier classifier. All classifiers use scikit-learn or LightGBM implementations. Random seeds are fixed for reproducibility; $n_{\text{jobs}} = 1$ for tree-based classifiers to ensure deterministic results. Values marked with a dagger differ from the library default of the corresponding implementation; all others are the defaults. The configurations are those of the released Hydra configs under `configs/experiment/`.

Table A1. Carrier classifier hyperparameters. [†] indicates a difference from the library default value, which is provided in parentheses.

Carrier	Family	Key Hyperparameters
Logistic Regression	Linear	$\text{max_iter} = 1000$ [†] [100], $\text{solver} = \text{lbfgs}$
MLP-shallow	Neural	hidden layers (64, 32) [†] [(100,)], $\text{activation} = \text{ReLU}$, $\text{solver} = \text{Adam}$, $\alpha = 10^{-4}$, $\text{batch size} = 256$ [†] [auto], $\text{max_iter} = 200$, $\text{early stopping} = \text{true}$ [†] [false] (validation fraction = 0.1)
LightGBM	Tree (boosted)	$n_{\text{estimators}} = 300$ [†] [100], $\text{learning rate} = 0.05$ [†] [0.1], $\text{max_depth} = 6$ [†] [−1, unbounded], $\text{num_leaves} = 31$
ExtraTrees	Tree (bagged)	$n_{\text{estimators}} = 100$, $\text{max_depth} = \text{None}$ (unbounded), $\text{min_samples_split} = 2$, $\text{criterion} = \text{entropy}$ [†] [gini]
Random Forest	Tree (bagged)	$n_{\text{estimators}} = 100$, $\text{max_depth} = \text{None}$ (unbounded), $\text{min_samples_split} = 2$, $\text{criterion} = \text{entropy}$ [†] [gini]

Two of these choices bear directly on the comparison between carriers. The depth cap on LightGBM is a deliberate choice, not a library default, as LightGBM grows trees without a depth limit unless one is set. A cap of 6 was imposed to keep the boosted ensemble within a regime where each tree is a weak learner. By contrast, the unbounded depth used for random forest and ExtraTrees is the scikit-learn default. The depth sweep in Section 4.4 and the uncapped LightGBM control reported there exist precisely to avoid this asymmetry being confounded with the carrier families themselves. The control alters two settings: it removes the cap and raises `num_leaves` from the default 31 to 255, since an unbounded depth at 31 leaves would leave the size of each tree unchanged.

Appendix B. Synthetic Perturbation Formulas

Table A2 details the five perturbation families used to construct the S0 benchmark shift. The headline composite applies families 1 and 3 simultaneously at intensity = 1.0.

Table A2. Synthetic perturbation families. * indicates that while the NSB family does touch WIDTH and LENGTH in the code, the resulting change in their standardised spread is below 0.001, which is an order of magnitude smaller than its effect on the two concentration features; see Appendix C.

#	Family	Headline Effect	Features Affected
1	NSB injection	Additive noise, scaling as the square root of SIZE	SIZE, CONC, CONC1, WIDTH *, LENGTH *
2	Flat-fielding uncertainty	Multiplicative Gaussian factor, 5% relative width	SIZE only
3	Atmospheric attenuation	Global factor 0.80 on SIZE	SIZE (−20%), WIDTH (−8%), LENGTH (−4%)
4	Zenith SIZE shift	Size-dependent attenuation	SIZE, LENGTH (+15%), WIDTH (+5%)
5	Energy threshold shift	Selective event rejection	Population size (events removed)

Appendix C. Per-Feature Effect of the Shifts

For each perturbation family used in this paper, Table A3 reports the change in mean and standard deviation of every affected feature, measured in the standardised space in which the alignment methods operate (mean over five seeds). The five features not listed (ASYM, M3LONG, DIST, ALPHA, M3TRANS) are untouched by all three shifts.

The table provides the basis for two precise statements in the main text. First, the atmospheric attenuation family is a pure translation only for SIZE, which the $\log_{10}$ transform converts from a multiplicative rescaling into an additive offset ($\Delta\sigma = 0.000$); for WIDTH and LENGTH, which are not log-transformed, the same rescaling moves the standard deviation as well as the mean, so the family is location-dominated rather than purely translational. Second, the NSB family inflates the spread of the two concentration features while barely moving any mean.

Table A3. Change in standardised mean and standard deviation, source to target, per feature and per shift family. Norm of the mean-shift vector: headline composite 0.1907, location only 0.1911, noise only 0.0184.

Feature	Headline Composite		Location Only (Atmospheric)		Noise Only (NSB)	
	$\Delta\mu$	$\Delta\sigma$	$\Delta\mu$	$\Delta\sigma$	$\Delta\mu$	$\Delta\sigma$
LENGTH	−0.0490	−0.0394	−0.0511	−0.0400	+0.0023	+0.0007
WIDTH	−0.0954	−0.0797	−0.0997	−0.0800	+0.0046	+0.0003
SIZE	−0.1567	+0.0028	−0.1548	−0.0000	−0.0019	+0.0028
CONC	+0.0107	+0.0273	+0.0000	+0.0000	+0.0107	+0.0273
CONC1	+0.0140	+0.0264	+0.0000	+0.0000	+0.0140	+0.0264

References

- Pagliaro, A.; Cusumano, G.; La Barbera, A.; La Parola, V.; Lombardi, S. Application of machine learning ensemble methods to ASTRI Mini-Array Cherenkov event reconstruction. *Appl. Sci.* **2023**, *13*, 8172. <https://doi.org/10.3390/app13148172>.
- Pagliaro, A.; La Barbera, A. Machine Learning for Event Reconstruction in Imaging Atmospheric Cherenkov Telescopes. In *Machine Learning Techniques for Astrophysics and Cosmology*; Bambi, C.; Kashyap, V.; Shashank, S.; Yoshida, N., Eds.; Springer: Singapore, 2026.
- Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>.
- Sun, B.; Feng, J.; Saenko, K. Return of Frustratingly Easy Domain Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016. <https://doi.org/10.48550/arXiv.1511.05547>.
- Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
- Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.C.; Vaughan, J.W. A Theory of Learning from Different Domains. *Mach. Learn.* **2010**, *79*, 151–175. <https://doi.org/10.1007/s10994-009-5152-4>.

7. Kouw, W.M.; Loog, M. A Review of Domain Adaptation without Target Labels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 766–785. <https://doi.org/10.1109/TPAMI.2019.2945942>.
8. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. <https://doi.org/10.1023/A:1010933404324>.
9. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. <https://doi.org/10.1007/s10994-006-6226-1>.
10. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 3146–3154.
11. Hillas, A.M. Cerenkov Light Images of EAS Produced by Primary Gamma Rays and by Nuclei. In *Proceedings of the 19th International Cosmic Ray Conference*, La Jolla, CA, USA, 11–23 August 1985; Volume 3, pp. 445–448.
12. CTA Observatory and CTA Consortium. CTAO Monte Carlo Simulations—Eventlists on DL2 Data Level—prod5. Zenodo, 2022. Dataset. Available online: <https://zenodo.org/records/7298569> (accessed on 6 October 2026).
13. Gueta, O. The Cherenkov Telescope Array: Layout, Design and Performance. In *Proceedings of the 37th International Cosmic Ray Conference (ICRC 2021)*, Virtual, 12–23 July 2021.
14. Kapoor, S.; Narayanan, A. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns* **2023**, *4*, 100804. <https://doi.org/10.1016/j.patter.2023.100804>.
15. Gulrajani, I.; Lopez-Paz, D. In Search of Lost Domain Generalization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual, 3–7 May 2021.
16. Musgrave, K.; Belongie, S.; Lim, S.N. Unsupervised Domain Adaptation: A Reality Check. *arXiv* **2021**, arXiv:2111.15672. <https://doi.org/10.48550/arXiv.2111.15672>.
17. Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. A Closer Look at Memorization in Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 233–242.
18. Wyner, A.J.; Olson, M.; Bleich, J.; Mease, D. Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers. *J. Mach. Learn. Res.* **2017**, *18*, 1–33.
19. Zhao, H.; Tachet des Combes, R.; Zhang, K.; Gordon, G.J. On Learning Invariant Representations for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 7523–7532.
20. Kantchelian, A.; Tygar, J.D.; Joseph, A.D. Evasion and Hardening of Tree Ensemble Classifiers. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York City, NY, USA, 19–24 June 2016; Volume 48, pp. 2387–2396.
21. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 2208–2217.
22. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
23. La Parola, V.; Cusumano, G.; Lombardi, S.; Compagnino, A.A.; La Barbera, A.; Tutone, A.; Pagliaro, A. Machine learning-enhanced discrimination of gamma-ray and hadron events using temporal features: an ASTRI Mini-Array analysis. *Appl. Sci.* **2025**, *15*, 3879. <https://doi.org/10.3390/app15073879>.
24. Li, T.P.; Ma, Y.Q. Analysis methods for results in gamma-ray astronomy. *Astrophys. J.* **1983**, *272*, 317–324. <https://doi.org/10.1086/161295>.

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.