MAJORITY IS THE REFERENCE YOU NEVER HAD

Anonymous authors
Paper under double-blind review

ABSTRACT

Feature shifts across data sources are prevalent in applications involving healthcare, biomedicine, socioeconomic, financial, surveys, and multi-sensor data. Such shifts can arise from unharmonized heterogeneous data sources, noisy measurements, or inconsistent processing and standardization pipelines, resulting in erroneous features. Localizing shifted features is essential for identifying the underlying causes and correcting or filtering affected data before they degrade downstream analyses. Existing methods typically rely on a prespecified trusted reference to identify feature shifts in query data. In practice, however, a fully trustworthy reference source is often unavailable, making conventional pairwise comparisons unable to determine which data source is responsible for an observed shift. We propose Multi-source Reference-Free Feature Shift Localization (MuRF-FSL), an end-to-end neural network that simultaneously localizes shifted features across multiple data sources without requiring an external clean reference. Under the assumption that, for each feature, a strict majority of sources remain normal, MuRF-FSL treats the pattern shared by this majority as the missing reference and learns to attribute shifts by combining source-specific statistical representations with cross-source deviation patterns. Trained on a large and diverse collection of datasets with simulated shifts, MuRF-FSL generalizes to previously unseen datasets and shift scenarios without retraining. Experiments show that, despite operating in the more challenging reference-free setting, MuRF-FSL achieves state-of-the-art localization performance with highly efficient inference and outperforms reference-based methods provided with clean reference data.

1 INTRODUCTION

Feature distribution shifts are widespread across data sources in many real-world applications involving multi-dimensional data which have a set of "corrupted" features. A feature distribution shift occurs when some features exhibit inconsistent statistical properties across data sources. In healthcare, biomedicine, and the life sciences, differences in experimental technologies, hardware devices, and data-processing pipelines across institutions may introduce source-specific structural biases. In clinical genomics, feature shifts may also arise from heterogeneous data-collection procedures, such as differences in genotyping array platforms or phenotype curation pipelines (Moreno-Grau et al., 2024). In the social sciences, streaming media, and e-commerce, combining tabular and structured data from multiple sources, regions, or time periods without proper homogenization steps can lead to mismatching and biased features due to incorrect data collection procedures, human entry errors, faulty standardization, or erroneous data processing (Barchard & Pace, 2011; Dai et al., 2015). Similarly, multi-sensor applications in the manufacturing industry, medicinal monitoring, finance analysis, and defense can suffer feature shifts due to faulty sensors and measuring devices (Qian et al., 2022; Barrabés et al., 2023). Although data pre-processing and homogenization can reduce such shifts, selecting an appropriate homogenization method is often difficult because it depends heavily on the specific data and relevant domain expertise (Lim et al., 2018). Unattended feature shifts that remain after unsuccessful homogenization can undermine the reliability of downstream analyses, leading to erroneous scientific conclusions or systematically biased machine learning models. Feature shift localization is therefore critical in many data-driven domains.

Feature shift localization aims to identify the specific features responsible for distributional shifts across data sources, thereby supporting the diagnosis of shift causes and the correction or filtering of affected data. Early methods focused on testing whether two datasets were drawn from the same

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

distribution, and most could not further identify which features had shifted, a task that belongs to distribution shift detection (Barrabés et al., 2025). Some recent feature shift localization techniques are promising and have improved localization accuracy, scalability to large and high-dimensional datasets, and generalization across datasets. However, they typically formulate the task as a pairwise comparison between a reference and a query distribution (Kulinski et al., 2020; Barrabés et al., 2023; 2025). These methods require a data source representing the normal state to be specified in advance, introducing a directionality ambiguity that cannot be resolved from pairwise differences alone. Kulinski et al. (2020) noted that, without prior knowledge, even when only one of two sensors is known to be faulty, it is impossible to determine which sensor is responsible. Consequently, pairwise comparison alone cannot reliably confirm that a source is entirely normal. Although comparing multiple sources against one another may help identify such a source, different sources may be shifted in different features, so in all likelihood such a source does not exist. This raises a central question: how can an effective comparison baseline be established and shifted features localized when no trusted reference source can be specified?

To address this problem, we propose the Multi-source Reference-Free Feature Shift Localization (MuRF-FSL) network. MuRF-FSL extends conventional pairwise comparison between two sources to joint inference across multiple sources. Under the assumption that each feature remains normal in a strict majority of sources, it treats the pattern shared by the majority for that feature as an internal comparison baseline. Specifically, MuRF-FSL first applies a shared statistical descriptor network to each data source to obtain per-feature statistical representations. It then constructs a source-specific internal reference through robust leave-one-source-out consensus and computes the corresponding source-relative robust residual (SR). An auxiliary enhancement module captures cross-source statistical variation, local inter-feature relationships, and consensus reliability. Finally, a shared prediction network combines these representations to produce a shift probability for every source-feature pair. MuRF-FSL is trained end to end on a large pool of datasets with diverse simulated feature shifts and evaluated on datasets and shift types unseen during training, demonstrating its ability to generalize without retraining. Designed for feature shift localization without a trusted reference source, it operates in a more challenging multi-source setting. Even in this setting, MuRF-FSL scales to large, high-dimensional datasets, remains invariant to sample order and equivariant to both source and feature order, and achieves the highest localization accuracy while maintaining efficient inference, outperforming reference-based methods granted access to clean reference data.

2 Related Work

Distribution Shift Detection and Feature Selection. Distribution shift detection determines whether two distributions differ, $p \neq q$, rather than which features account for the difference (Gama et al., 2014; Lu et al., 2018; Pan et al., 2020; Rabanser et al., 2019). The shifts considered include marginal, concept, covariate, and label shifts (Lu et al., 2016; Losing et al., 2016; Liu et al., 2020), and recent methods range from hypothesis testing for concept drift (Yu et al., 2018) to multivariate two-sample tests based on Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) and marginal Kolmogorov-Smirnov (KS) tests (Rabanser et al., 2019). The feature-selection methods considered here seek features relevant to classification or regression, providing interpretability and removing redundancy. Common approaches include wrapper, filter, and embedded methods (Maldonado & Weber, 2009; Mustaqeem et al., 2017; Nasir et al., 2020; Hopf & Reifenrath, 2021; Tran et al., 2016; Huang et al., 2018). Representative filter scores include Mutual Information (MI), ANOVA-F, the Chi-square test, MRMR, and FAST-CMIM (Battiti, 1994; Elssied et al., 2014; Bahassine et al., 2020; Ding & Peng, 2005; Li et al., 2017; Fleuret, 2004), while embedded methods use model-internal rankings such as logistic-regression coefficients or Gini importance (Cheng et al., 2006; Sylvester et al., 2018). Neither task addresses our problem: shift detection returns a distribution-level decision without localizing features, while these selection methods rank features for a predictive task rather than attribute source-specific distribution shifts.

Feature Shift Localization. Feature shift localization instead seeks to identify the individual features responsible for such a difference. Kulinski et al. (2020) introduced a notable conditional testing framework that accurately localizes shifted features through both model-free and model-based variants: K-Nearest Neighbors with the KS statistic (KNN-KS), a multivariate Gaussian model with KS tests (MB-KS), a multivariate Gaussian model with Fisher-divergence test statistics (MB-SM), and deep density models with a Fisher-divergence test (Deep-SM). DataFix (Barrabés et al., 2023) sub-

sequently improved localization accuracy by iteratively training a random forest to distinguish the reference distribution from the query distribution, removing features with the highest impurity-based importance scores until the divergence is minimized, and using a knee-detection algorithm to determine the optimal stopping point. Although DataFix performs well in many settings, it struggles with challenging feature shifts and scales poorly to large, high-dimensional datasets because its repeated random forest training incurs substantial computational overhead. FSL-Net (Barrabés et al., 2025) further applies a shared Statistical Descriptor Network to extract distribution-level representations and an end-to-end Prediction Network to produce feature-wise shift probabilities, improving both localization accuracy and scalability to large, high-dimensional datasets. Despite their differences in statistical tests, iterative procedures, and neural architectures, these methods all formulate localization as a pairwise comparison between a prespecified reference dataset and a query dataset, thereby assuming that the reference represents the normal state. However, an observed pairwise discrepancy alone cannot determine whether the shift originates from the query or the reference, nor can this formulation directly accommodate the demanding reference-free settings in which localization is most urgently needed.

Sparse Contamination and the Majority-Normal Assumption. Treating anomalies as sparse deviations from a dominant pattern is a common principle in anomaly detection and robust statistics, where anomalies are few and differ from a recurring normal pattern (Liu et al., 2012). Cellwise contamination models are especially relevant, since they allow a small number of corrupted features per observation and therefore require no observation to be entirely clean (Alqallaf et al., 2009; Rousseeuw & Bossche, 2018). Conventional high-breakdown multivariate estimators may nonetheless fail because cellwise outliers can propagate in high dimensions (Alqallaf et al., 2009). Redundant sensing can distinguish localized sensor malfunctions from process upsets (Stork & Kowalski, 1999), while group testing can detect rare faulty sensors (Lo et al., 2015). Cataltepe et al. (2005) use the median of estimates from other correlated sensors to identify faults when fewer than half are faulty. Motivated by these settings, we adopt a feature-wise multi-source analogue: for every feature, a strict majority of sources are assumed to remain normal, without requiring any individual source to be normal across all features. Coordinated shifts affecting a majority of sources on the same feature fall outside this assumption.

Reference-Free Multi-Source Inference. Multi-source inference without a prespecified clean reference takes several forms. Multi-sample tests such as DISCO assess homogeneity across distributions but provide an overall decision rather than source-wise and feature-wise shift attribution (Rizzo & Székely, 2010). Truth-discovery methods instead infer latent truths and source reliability from conflicting reports without external ground truth (Yin et al., 2007; Li et al., 2016). In redundant sensing, Saha et al. (2023) combine cross-sensor agreement with historical credibility to estimate a common signal. When credibility information is unavailable, agreement among a reliable majority can provide an internal baseline. A related robustness principle appears in distributed learning, where Yin et al. (2018) use coordinate-wise medians and trimmed means to aggregate worker gradients despite a minority of Byzantine participants under their stated assumptions. These studies show how multiple sources can support testing or estimation without a designated clean reference. Their outputs, however, are overall test decisions, estimated truths or signals, aggregate gradients, or source reliability scores, rather than an attribution of shifted features to individual sources.

Permutation Symmetry in Multi-Source Modeling. Permutation invariance and equivariance describe whether a model's output remains unchanged or transforms consistently when its inputs are reordered (Lim & Nelson, 2022). Deep Sets obtains permutation-invariant representations through shared transformations and symmetric aggregation (Zaheer et al., 2017). Set Transformer uses attention to model interactions among set elements while retaining permutation-invariant outputs (Lee et al., 2019). Training with transformed inputs can encourage learned invariance without guaranteeing exact symmetry (Benton et al., 2020). In feature shift localization, FSL-Net (Barrabés et al., 2025) obtains sample-order invariance through symmetric statistics and pooling, and uses feature shuffling to encourage approximate feature-order equivariance. MuRF-FSL retains sample-order invariance through its shared descriptor network, achieves source-order equivariance through shared source processing and symmetric leave-one-source-out consensus, and uses randomized feature ordering to encourage approximate feature-order equivariance. Accordingly, reordering sources reorders the corresponding prediction rows, whereas reordering samples leaves the predictions unchanged.

3 MULTI-SOURCE REFERENCE-FREE FEATURE SHIFT LOCALIZATION

Problem Formalization. We generalize the conventional empirical feature shift localization task (Barrabés et al., 2025) to a multi-source setting in which no trusted reference source exists.

Definition 1. [Empirical Multi-Source Reference-Free Feature Shift Localization Task]. Given $K \geq 3$ data sources, where source $k \in [K]$ provides a set of d-dimensional samples $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{N_k}$ drawn from distribution $q^{(k)}$ with $\boldsymbol{x}_i^{(k)} \in \mathbb{R}^d$, and let p denote the shared d-dimensional distribution that all sources follow before any source-specific feature shift occurs, where $[K] = \{1, \ldots, K\}$. The empirical multi-source reference-free feature shift localization task consists of predicting, through a mapping F, the set of shifted features from the K observed datasets, i.e., $\big(\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(K)}\big) = F\big(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\big)$, such that for every $k \in [K]$ it holds that $D\Big(q^{(k)}_{\overline{\mathcal{C}}^{(k)}}, p_{\overline{\mathcal{C}}^{(k)}}\Big) = 0$, $D\big(q^{(k)}, p\big) > 0$, and $\mathcal{C}^{(k)} = \underset{D(q^{(k)}_{\overline{\mathcal{C}}^{(k)}}, p_{\overline{\mathcal{C}}^{(k)}})=0}{\arg\min} |\mathcal{C}^{(k)}|$, where D is some valid statistical distance.

For source k, if removing the shifted features $\mathcal{C}^{(k)}$ and retaining only the unshifted features $\mathcal{S}^{(k)} = \overline{\mathcal{C}}^{(k)}$ yields $D(q^{(k)}_{\mathcal{S}^{(k)}}, p_{\mathcal{S}^{(k)}}) = 0$, then a feature shift caused by $\mathcal{C}^{(k)}$ is said to exist between the distributions p and $q^{(k)}$. We refer to $\mathcal{X}^{(k)}$ as the query set of source k. The sources may have different sample sizes, but share d semantically aligned and consistently ordered features. Furthermore, we call $q^{(k)}$ its observed distribution and p the normal distribution. The latter serves only to define the task. In practice, all $q^{(k)}$ are unknown and can only be observed through finite samples. Therefore, an empirical mapping $\widetilde{F}$ is needed to approximate the ideal mapping F, predicting for each source k a set $\widehat{\mathcal{C}}^{(k)}$ that is as close as possible to the true shifted feature set $\mathcal{C}^{(k)}$, i.e., $\Big(\widehat{\mathcal{C}}^{(1)}, \ldots, \widehat{\mathcal{C}}^{(K)}\Big) = \widetilde{F}\big(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\big)$.

The above definition encompasses the following representative shift types: *marginal shift*, where there exists $j \in \mathcal{C}^{(k)}$ such that $D(q^{(k)}_j, p_j) > 0$, i.e., the univariate marginal distribution of at least one shifted feature changes. *Dependence shift* satisfies $D(q^{(k)}, p) > 0$ but $D(q^{(k)}_j, p_j) = 0$ for all $j \in [d]$, i.e., all univariate marginal distributions remain identical while the multivariate dependence structure changes. *Cross-subset dependence shift* is a special case of dependence shift satisfying $D(q^{(k)}_{\mathcal{S}^{(k)}}, p_{\mathcal{S}^{(k)}}) = 0$, $D(q^{(k)}_{\mathcal{C}^{(k)}}, p_{\mathcal{C}^{(k)}}) = 0$, and $D(q^{(k)}, p) > 0$. In this case, the internal joint distributions of the shifted and unshifted feature sets each remain unchanged, and only the dependence between the two sets changes.

Assumptions and Identifiability. Because the latent normal distribution p is unobservable, multiple symmetrically positioned sources alone cannot determine which pattern of differences represents the normal state. To constrain the normal baseline in the reference-free setting and make the attribution of shifts identifiable, we adopt the following two assumptions.

Within-source partial shift assumption. We focus on the setting in which every input source contains only a partially shifted set of features, i.e., $\varnothing \neq \mathcal{C}^{(k)} \subsetneq [d]$ for all $k \in [K]$.

This setting does not provide a fully normal source that can directly serve as a reference, while still guaranteeing that each source retains at least one unshifted feature. If all features of a source were shifted, then, in the absence of external reference data, it would be difficult to determine whether the overall distributional difference stems from natural inter-source variation or from feature shifts, so we do not consider this case.

Per-feature strict-majority normal assumption. Let $C_{k,j} = \mathbb{I}(j \in \mathcal{C}^{(k)})$ indicate whether feature j of source k is shifted. For every feature $j \in [d]$, we require $\sum_{k=1}^{K} C_{k,j} \leq \big\lfloor \frac{K-1}{2} \big\rfloor$.

This condition guarantees that each feature remains normal in a strict majority of sources, so that the cross-source majority pattern can serve as the internal normal reference for that feature. It only constrains the number of sources in which the same feature is shifted, and does not require the existence of a source that is fully normal across all features. Hence, different sources may exhibit different shift types on different, partially overlapping feature subsets. Together, these two assump-

tions rule out the ambiguity between normal and shifted patterns that cannot be distinguished in the reference-free setting, and provide the identifiability basis for per-source, per-feature localization.

Overview of MuRF-FSL. The proposed MuRF-FSL is an end-to-end trained multi-source reference-free feature shift localization network that simultaneously predicts the set of shifted features across the K data sources: $\left(\widehat{\mathcal{C}}^{(1)}, \ldots, \widehat{\mathcal{C}}^{(K)}\right) = \widetilde{F}_\Theta\big(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\big)$. MuRF-FSL infers the probability that each feature of each source belongs to the corresponding shifted set. That is, given K symmetric datasets sharing the same feature space, the network predicts a $K \times d$ probability matrix $\widehat{\boldsymbol{P}} = \Psi_\Theta\big(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\big)$, where the (k, j)-th entry denotes the probability that feature j of source k is shifted, i.e., $\widehat{P}(j \in \mathcal{C}^{(k)}) = \widehat{P}_{k,j} = \big[\Psi_\Theta\big(\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\big)\big]_{k,j}$. The full predicted shifted feature set is obtained by selecting the features whose probability is not below 0.5: $\widehat{\mathcal{C}}^{(k)} = \{j \in [d] : \widehat{P}_{k,j} \geq 0.5\}$. Unlike the conventional pairwise formulation, which takes a reference source and a query source as input, MuRF-FSL exploits the per-feature strict-majority-normal assumption to construct source-relative statistical references from the observed data and, like FSL-Net, operates without prior knowledge of the number of shifted features.

MuRF-FSL applies FSL-Net's pretrained shared **Statistical Descriptor Network (SD)** ϕ_θ (Appendix A) (Barrabés et al., 2025) to each source $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{N_k}$, yielding $\boldsymbol{H}^{(k)} = \phi_\theta(\mathcal{X}^{(k)}) \in \mathbb{R}^{d \times B}$ and their stack $\boldsymbol{H} \in \mathbb{R}^{K \times d \times B}$. It constructs a **Source-Relative Robust Residual (SR)** $\boldsymbol{R}^{(k)}$ against a leave-one-source-out consensus. An optional **Auxiliary Enhancement Module** η_ω combines the Cross-Source Statistical Moment (CSM), Cross-Source Neural (CSN), and Consensus Stability and Uncertainty (CSU) branches as $\boldsymbol{U}_{\text{aux}}$ (Appendix B). The fused descriptor $\boldsymbol{Z} = [\boldsymbol{H}; \boldsymbol{R}; \boldsymbol{U}_{\text{aux}}]$ is transformed into $\widetilde{\boldsymbol{Z}}$ and mapped by a shared **Prediction Network (PN)** γ_ψ to $\widehat{\boldsymbol{P}} = \gamma_\psi(\widetilde{\boldsymbol{Z}})$. This architecture handles varying sample sizes, feature dimensions, and source counts without retraining.

Source-Relative Robust Residual. SR operationalizes the per-feature strict-majority-normal assumption: although no source is entirely clean, each feature's normal pattern is shared by a strict majority of sources. A leave-one-out peer median provides an estimated internal reference without allowing the target source to influence its own baseline. The residual measures its deviation from that reference rather than from a prespecified clean dataset. Given $\boldsymbol{H} \in \mathbb{R}^{K \times d \times B}$, SR constructs a residual tensor $\boldsymbol{R} \in \mathbb{R}^{K \times d \times B}$. Let $\boldsymbol{h}_{k,j} = \boldsymbol{H}_{k,j,:} \in \mathbb{R}^B$ and $\mathcal{K}_{-k} = [K] \setminus \{k\}$. For every source k and feature j, we compute the leave-one-out median $\boldsymbol{m}_j^{(-k)} = \text{median}_{\ell \in \mathcal{K}_{-k}} \boldsymbol{h}_{\ell,j}$, robust scale $\boldsymbol{s}_j^{(-k)} = \max\Big(c_{\text{MAD}}\, \text{median}_{\ell \in \mathcal{K}_{-k}} \Big|\boldsymbol{h}_{\ell,j} - \boldsymbol{m}_j^{(-k)}\Big|, s_{\min}\Big)$, mean $\overline{\boldsymbol{h}}_j^{(-k)} = \frac{1}{K-1} \sum_{\ell \in \mathcal{K}_{-k}} \boldsymbol{h}_{\ell,j}$, and standard deviation $\boldsymbol{\sigma}_j^{(-k)} = \sqrt{\max\left(\frac{1}{K-1} \sum_{\ell \in \mathcal{K}_{-k}} \boldsymbol{h}_{\ell,j}^2 - \left(\overline{\boldsymbol{h}}_j^{(-k)}\right)^2, \varepsilon\right)}$, where all operations are applied elementwise over the descriptor channels, $c_{\text{MAD}} = 1/\Phi^{-1}(0.75) \approx 1.4826$, and $s_{\min} > 0$ and $\varepsilon > 0$. Gradients through $\boldsymbol{m}_j^{(-k)}$ and $\boldsymbol{s}_j^{(-k)}$ are stopped during backpropagation. The standardized residual $\boldsymbol{r}_j^{(k)} = \left(\boldsymbol{h}_{k,j} - \boldsymbol{m}_j^{(-k)}\right) / \boldsymbol{s}_j^{(-k)}$ is compressed as $\boldsymbol{R}_j^{(k)} = \text{clip}\Big(\text{slog}\Big(\boldsymbol{r}_j^{(k)}\Big), -r_{\max}, r_{\max}\Big)$, where $\text{slog}(x) = \text{sign}(x) \log(1 + |x|)$. Stacking all $\boldsymbol{R}_j^{(k)}$ yields $\boldsymbol{R}$, while $\boldsymbol{m}$, $\boldsymbol{s}$, $\overline{\boldsymbol{h}}$, and $\boldsymbol{\sigma}$ are retained for auxiliary enhancement.

Auxiliary Enhancement Module. The auxiliary enhancement module groups CSM, CSN, and CSU and maps $\boldsymbol{H}$, $\boldsymbol{R}$, and the SR reference statistics to $\boldsymbol{U}_{\text{aux}} = \eta_\omega(\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{m}, \boldsymbol{s}, \overline{\boldsymbol{h}}, \boldsymbol{\sigma}) \in \mathbb{R}^{K \times d \times D_a}$. These additional descriptor channels complement the base descriptor and SR with cross-source variation, local feature relations, and consensus-reliability information. Its internal branches are detailed only in Appendix B. The fused descriptor is $\boldsymbol{Z} = [\boldsymbol{H}; \boldsymbol{R}; \boldsymbol{U}_{\text{aux}}] \in \mathbb{R}^{K \times d \times D}$, where $D = 2B + D_a$.

Prediction Network. The prediction network γ_ψ converts the transformed prediction input $\widetilde{\boldsymbol{Z}} \in \mathbb{R}^{K \times d \times D}$ into per-source, per-feature shift probabilities $\widehat{\boldsymbol{P}} \in [0, 1]^{K \times d}$. It follows the design of FSL-Net in making local predictions along the feature axis, but no longer compares a pre-specified reference descriptor with a query descriptor. Cross-source comparison has already been performed by SR and the auxiliary enhancement module, so the prediction network directly learns the mapping from this source-relative input to shift probabilities. The base descriptors contain explicit and

learned statistics of differing magnitudes. To prevent the larger values among them from dominating the subsequent convolutions and their gradients, a signed logarithmic transform is applied before prediction to the base-descriptor block of $\boldsymbol{Z}$ only, while the remaining channels, already robustly compressed or normalized, are left unchanged. For source k, the actual input to the prediction network is $\widetilde{\boldsymbol{Z}}^{(k)} = \left[\mathrm{slog}(\boldsymbol{H}^{(k)}); \boldsymbol{R}^{(k)}; \boldsymbol{U}^{(k)}_{\mathrm{aux}}\right] \in \mathbb{R}^{d\times D}$. The descriptor channels are treated as the input channels of one-dimensional convolutions and the feature dimension as the sequence axis. The input first passes through a projection layer with kernel size 1, group normalization, and a nonlinear activation, yielding $\boldsymbol{Y}_0^{(k)} = \mathrm{GELU}(\mathrm{GN}(\mathrm{Conv}_1(\widetilde{\boldsymbol{Z}}^{(k)\top}))) \in \mathbb{R}^{D_p\times d}$, where D_p is the number of hidden channels. Afterwards, $\boldsymbol{Y}_0^{(k)}$ passes through L_p shared residual convolution blocks along the feature axis. The l-th block is $\boldsymbol{V}_l^{(k)} = \mathrm{GELU}(\mathrm{GN}_{l,1}(\mathrm{Conv}_\kappa(\boldsymbol{Y}_{l-1}^{(k)})))$ and $\boldsymbol{Y}_l^{(k)} = \mathrm{GELU}(\mathrm{GN}_{l,2}(\mathrm{Conv}_\kappa(\boldsymbol{V}_l^{(k)})) + \boldsymbol{Y}_{l-1}^{(k)})$ for $l = 1,\dots,L_p$, where κ is the convolution kernel size. Length-preserving padding maintains the correspondence between input features and output positions. A final convolution produces logits $\boldsymbol{L}^{(k)} = \mathrm{Conv}_{\mathrm{out}}(\boldsymbol{Y}_{L_p}^{(k)}) \in \mathbb{R}^d$, followed by $\widehat{\boldsymbol{P}}^{(k)} = \mathrm{sigmoid}(\boldsymbol{L}^{(k)})$. The same predictor processes all sources with shared parameters, so permuting the input sources only permutes the source axis of the output. Given threshold τ, the final decisions are $\widehat{C}_{k,j} = \mathbb{I}(\widehat{P}_{k,j} \ge \tau)$ without requiring the true shift fraction or number of shifted features.

Loss Function. MuRF-FSL jointly optimizes supervised source-feature classification and the geometry of the fused descriptor $\boldsymbol{Z}$. Given the ground-truth matrix $\boldsymbol{C} \in \{0,1\}^{K\times d}$, logits $\boldsymbol{L} \in \mathbb{R}^{K\times d}$, and $\widehat{\boldsymbol{P}} = \mathrm{sigmoid}(\boldsymbol{L})$, the classification loss is $\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{Kd}\sum_{k=1}^{K}\sum_{j=1}^{d}\left[C_{k,j}\log\widehat{P}_{k,j} + (1 - C_{k,j})\log(1-\widehat{P}_{k,j})\right]$, evaluated directly from $\boldsymbol{L}$ for numerical stability. To structure same-feature representations across sources, we apply a pairwise distance loss to the active blocks of $\boldsymbol{Z}$. By default, $\mathcal{G} = \{\mathrm{base}, \mathrm{res}, \mathrm{mom}, \mathrm{n}, \mathrm{unc}\}$, where base and res denote $\boldsymbol{H}$ and $\boldsymbol{R}$, and mom, n, and unc denote the optional CSM, CSN, and CSU branches. Disabling an auxiliary branch removes its block from both $\boldsymbol{Z}$ and $\mathcal{G}$, so the loss is computed over the remaining blocks. For block g of width D_g and channel c, we compute $s_c^{[g]} = \max(\mathrm{Std}_{k,j}(Z^{[g]}_{k,j,c}), \varepsilon_s)$ and $\overline{Z}^{[g]}_{k,j,c} = Z^{[g]}_{k,j,c}/s_c^{[g]}$, with gradients through $s_c^{[g]}$ stopped. For feature j of sources a and b, the block-balanced squared distance is $\Delta_{a,b,j} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\frac{1}{D_g}\sum_{c=1}^{D_g}\left(\overline{Z}^{[g]}_{a,j,c} - \overline{Z}^{[g]}_{b,j,c}\right)^2$. Using shift labels and training manipulation identities, we define $\Omega_{00} = \{(a,b,j) : a<b,\, C_{a,j} = C_{b,j} = 0\}$, $\Omega_{01} = \{(a,b,j) : a < b,\, C_{a,j} \ne C_{b,j}\}$, and $\Omega_{11}^{\neq} = \{(a,b,j) : a<b,\, C_{a,j} = C_{b,j} = 1,\, O_{a,j} \ne O_{b,j}\}$, where $O_{k,j}$ is used only to construct training pairs. The auxiliary objective is $\mathcal{L}_{\mathrm{pair}} = \mathrm{mean}_{(a,b,j)\in\Omega_{00}}\Delta_{a,b,j} + \mathrm{mean}_{(a,b,j)\in\Omega_{01}}[m_{01} - \Delta_{a,b,j}]_+ + \beta\,\mathrm{mean}_{(a,b,j)\in\Omega_{11}^{\neq}}[m_{11} - \Delta_{a,b,j}]_+$, where an empty-set term is zero and $[x]_+ = \max(x,0)$. Here, $\mathcal{L}_{\mathrm{BCE}}$ supervises shift decisions, while $\mathcal{L}_{\mathrm{pair}}$ draws unshifted representations together and separates pairs differing in shift status or manipulation. The final objective is $\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{pair}}\mathcal{L}_{\mathrm{pair}}$.

Permutation Properties. MuRF-FSL is invariant to sample order because its statistical functionals sort samples and its learned descriptors use sample-mean pooling (Barrabés et al., 2025). Since all subsequent modules operate on these descriptors, reordering samples also leaves the final shift probabilities unchanged. Feature-order equivariance concerns applying the same permutation of the d feature columns to every source, preserving cross-source feature correspondence. Marginal statistics and residual paths preserve feature correspondence, but feature-axis convolutions preclude exact feature-order equivariance. Randomized feature selection and ordering during training encourage approximate equivariance. This randomization cannot guarantee exact equivariance, since reordering features also changes the local neighborhoods seen by the convolutions. Source-order equivariance, by contrast, holds exactly: shared modules process each source identically, and the leave-one-source-out consensus and auxiliary cross-source statistics are symmetric in source indices. Writing $\mathcal{X} = (\mathcal{X}^{(1)},\dots,\mathcal{X}^{(K)})$, for any permutation π of the sources, $\widehat{\boldsymbol{P}}(\pi\mathcal{X}) = \pi\widehat{\boldsymbol{P}}(\mathcal{X})$: the median and scale computed from each leave-one-out source set follow the permuted source, and the shared predictor preserves this correspondence. Both loss terms are also symmetric over source indices, so training does not privilege a particular source position.

Training and Validation Data and Shift Simulation. Training and validation use 2,610 and 129 disjoint matrices. Each episode shuffles rows and features, samples 5,000–20,000 rows and $d' \in$

$[8, 256]$ features (up to 512 for validation), normalizes them feature-wise to $[0, 1]$, and partitions them without replacement into $K \in \{3, \ldots, 10\}$ observable sources plus an auxiliary pool. An episode-level shift fraction $\rho \in \{0.05, 0.10, 0.25\}$ and source-specific, type-compatible adaptations of FSL-Net's T1–T8 (Barrabés et al., 2025) are sampled uniformly under strict-majority normality. Training and validation share these T operations, which differ from benchmark E1–E10. Appendix C details data preparation, episode construction, and the T1–T8 manipulations.

4 Experimental Results

Evaluation Setup. We adapt the evaluation manipulation taxonomy of Barrabés et al. (2023) to the multi-source setting and use the same open-source datasets as Barrabés et al. (2025), supplemented with one categorical dataset to broaden categorical coverage. The resulting datasets vary in data type, feature dimensionality, ranging from 8 to $10{,}000$, and sample size, ranging from $2{,}407$ to $70{,}000$. Table 1 summarizes each dataset, including its type, number of samples, and feature dimensionality. The continuous datasets are drawn from the UCI repository, including Gas (Huerta et al., 2016), Energy (Candanedo et al., 2017), and Musk2 (Blake, 1998), and from OpenML, including Scene (Boutell et al., 2004), MNIST (Deng, 2012), and Dilbert (Vanschoren et al., 2014). They further include the *United States COVID-19 Cases and Deaths by State over Time* dataset released by the U.S. Centers for Disease Control and Prevention (CDC), together with two simulated datasets generated using algebraic functions different from those employed in our training and validation simulations (Cosine and Polynomial) (Barrabés et al., 2023). The categorical datasets comprise Founders, which contains binary-encoded human DNA sequences (Perera et al., 2022), and the additionally introduced Reuters21578 dataset, constructed from Reuters-21578 news articles (Lewis, 1997) with each document represented as a binary bag of words. Each dataset is normalized feature-wise to $[0, 1]$, and its samples are randomly partitioned into K observable sources and one independent clean source, where $K \in \{3, \ldots, 10\}$. The clean source is used only to construct test manipulations that require clean values and as the reference input to reference-based baselines. Following (Barrabés et al., 2023), a fraction of 5%, 10%, or 25%, denoted by ρ, of the features is manipulated. Ten manipulation types apply to continuous datasets and eight apply to categorical datasets, denoted E1–E10 and defined in Appendix D. For every observable source, $\max(1, \lfloor \rho d \rfloor)$ features are manipulated, while each feature is constrained to remain normal in a strict majority of sources. For every number of sources K and shift fraction, we generate as many episodes as there are applicable manipulation types. Within each episode, one applicable manipulation is sampled independently for each observable source. Each continuous dataset contains 240 episodes and each categorical dataset contains 192 episodes.

Baselines. We compare MuRF-FSL with five methods: three feature shift localization methods, namely DataFix, FSL-Net w/ Ref, and FSL-Net w/o Ref, and two feature selection methods, namely MI and SelectKBest. Each selected method offers an advantage in at least one of the two relevant dimensions, localization capability and inference speed. DataFix and FSL-Net w/ Ref represent the strongest existing localization methods, while MI and SelectKBest are tuned univariate methods identified by Barrabés et al. (2023) as having very low computational cost. FSL-Net w/o Ref provides a second reference-free comparison point and tests whether simply removing the reference input is sufficient to obtain reference-free localization. All five baselines and MuRF-FSL are evaluated on the same episodes. Implementation details for all baselines, including their reference-access assumptions, decision rules, and the construction of FSL-Net w/o Ref, are provided in Appendix E. Appendix F records their numerical settings, together with the model hyperparameters and the checkpoint selection procedure.

Evaluation Protocol. An *episode* denotes one multi-source localization instance comprising K observable datasets generated from a normalized base matrix through source-specific feature manipulations and their ground-truth shifted sets. It is the unit of training, evaluation, and runtime measurement and produces a $K \times d$ prediction matrix. Localization is measured by F1, and computational cost by the mean and maximum wall-clock runtime per episode. Except DataFix, all methods process episodes serially on one RTX 5090 GPU. DataFix remains CPU-based because replacing its random-forest estimator would change DF-Locate's feature-removal sequence and stopping point, thereby altering its semantics. In the benchmark of Barrabés et al. (2025), DataFix required roughly ten hours on high-dimensional datasets versus only minutes for FSL-Net. It is therefore excluded from our hardware-controlled runtime statistics and figures.

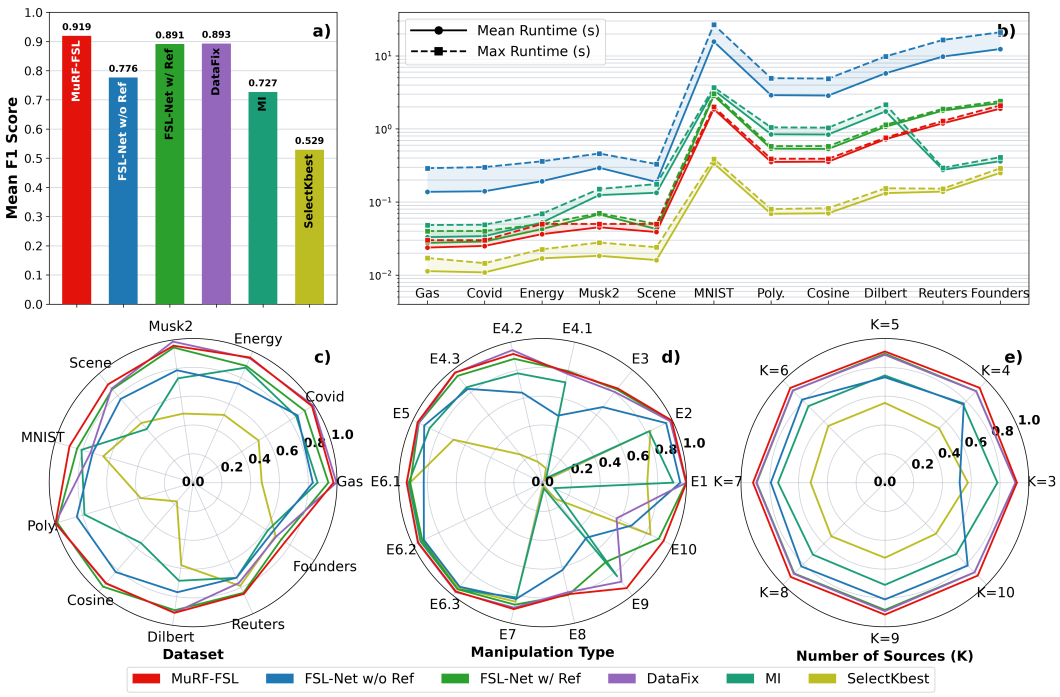


Figure 1: Performance and runtime comparison across feature shift localization methods: (a) dataset-equal mean F1, (b) mean and maximum runtime on each dataset, (c) mean F1 by dataset, (d) mean F1 by manipulation type, and (e) mean F1 by the number of observable sources K.

Feature Shift Localization Performance and Runtime. Figure 1 compares the localization performance of all methods and the runtime of methods evaluated under the common GPU setting. Figure 1a reports the dataset-equal mean F1, where each dataset-level F1 is first obtained by pooling predictions over all observable source counts K, shift fractions, and applicable manipulation types within that dataset. Figure 1b reports the mean and maximum per-episode runtime of each GPU-evaluated method on every dataset, characterizing typical and worst-case computational demands, respectively. MuRF-FSL achieves the highest mean F1 of 0.9192, outperforming the strongest reference-based localization method, DataFix, at 0.8928, and the second-best method, FSL-Net w/ Ref, at 0.8910. Although SelectKBest and MI are computationally efficient, their localization performance is clearly the weakest. MuRF-FSL maintains seconds-level inference on every dataset and is faster than FSL-Net w/ Ref in both mean and maximum runtime. To simulate the realistic absence of a trustworthy reference source, we construct the FSL-Net w/o Ref adaptation described above. Relative to FSL-Net w/ Ref, its F1 decreases by 0.1147, while its mean per-episode runtime increases by a factor of 4.3 to 5.5. These results show that MuRF-FSL can jointly process multiple sources without a clean, unshifted reference and scale to large, high-dimensional datasets.

Performance across Datasets, Manipulation Types, and Numbers of Sources. Figures 1c, 1d, and 1e report F1 by dataset, manipulation type, and number of observable sources, respectively. MuRF-FSL achieves the best result on six of the eleven datasets and eight of the fourteen manipulation conditions. Its advantage is consistent on high-dimensional real datasets such as MNIST, Dilbert, Reuters21578, and Founders, whereas reference-based methods retain an advantage on some continuous datasets. This indicates that performance differences depend not only on feature dimensionality but also on the dependency structure of the data. The manipulation-level results further support this interpretation. E3 and E8 preserve univariate marginal distributions while disrupting feature dependencies, causing MI and SelectKBest to fail almost completely. MuRF-FSL continues to identify these shifts effectively, indicating that its cross-source statistical descriptors and feature-relation modeling capture structural changes missed by univariate tests. MuRF-FSL also addresses the pronounced weakness of DataFix on E10. Its remaining weaknesses are concentrated on E4.1 and E4.2, where the signal is close to the noise level, and on E6.1–E6.3, which randomly flip cate-

Table 1: Datasets when benchmarking.

Dataset Type	Dataset	# Features	# Samples
Continuous	Gas	8	12,815
	Covid	10	9,889
	Energy	26	19,735
	Musk2	166	6,598
	Scene	294	2,407
	MNIST	784	70,000
	Polynomial	1,000	10,000
	Cosine	1,000	10,000
	Dilbert	2,000	10,000
Categorical	Reuters21578	1,227	21,578
	Founders	10,000	4,144

Table 2: Mean F1 by ablation configuration.

Model Configuration	Mean F1
SM	0.5728
SD + PN	0.6670
SD + SR + PN	0.9074
SD + SR + CSM + PN	0.9132
SD + SR + CSN + PN	0.9052
SD + SR + CSU + PN	0.9082
SD + SR + CSN + CSU + PN	0.9136
SD + SR + CSM + CSU + PN	0.9143
SD + SR + CSM + CSN + PN	0.9183
SD + SR + CSM + CSN + CSU + PN	**0.9192**

gorical values. In these cases, a clean reference provides a more direct comparison. As the number of sources increases from $K = 3$ to $K = 10$, MuRF-FSL consistently ranks first and exhibits only minor performance variation. It remains stable even at $K = 3$, when only two peer sources are available to construct the internal reference for each target source, whereas the reference-free FSL-Net adaptation degrades substantially. This result suggests that reliable majority consensus is more important than merely increasing the number of pairwise comparisons.

Ablation Analysis. Table 2 reports dataset-equal F1 across component ablations. SM denotes the explicit statistical-measure branch of SD, excluding its learned moment extraction and neural embedding branches. We report SM-only as the PN-free baseline because FSL-Net defines a direct thresholding rule for these explicit statistics, whereas its learned descriptor branches are trained jointly with PN. The SM-only row adapts FSL-Net's threshold-based statistical-measure baseline (Barrabés et al., 2025) to our reference-free setting. Appendix G details its scoring and validation-calibrated threshold. Its F1 is 0.5728, versus 0.6670 for SD+PN. Since both the descriptor channels and decision rule change, this contrast does not isolate the effect of PN. Adding SR to SD+PN raises F1 by 0.2404 to 0.9074, the largest gain in the table. This supports the role of the source-relative comparison with an internal consensus reference. The full auxiliary module raises F1 by a further 0.0118 to 0.9192. Every auxiliary row includes SR, so these comparisons test refinements to an already strong internal reference. Aggregate F1 does not show whether those gains concentrate in particular datasets or shift types. CSM alone adds 0.0058 to the SR configuration, whereas CSU alone adds only 0.0008. The branches do not help uniformly on their own: CSN alone lowers F1 to 0.9052, yet CSM+CSN reaches 0.9183, above CSM alone at 0.9132. CSN+CSU also exceeds either branch alone, whereas CSM+CSU improves by just 0.0011 over CSM. Together, these ablations support the central role of the source-relative internal reference, with auxiliary branches providing smaller, combination-dependent refinements.

5 Conclusion

MuRF-FSL is the first method for multi-source feature shift localization that requires no prespecified clean reference at inference time. Under the within-source partial-shift and per-feature strict-majority-normal assumptions, it uses shared statistical descriptors and robust leave-one-source-out consensus to construct an internal reference for each source. Source-relative residuals provide the main comparison signal, while optional auxiliary branches capture cross-source variation, local feature relations, and consensus reliability before a shared predictor produces a $K \times d$ matrix of shift probabilities. The model is invariant to sample order, equivariant to source order, and approximately equivariant to feature order. It processes varying source counts and feature dimensions without retraining. On eleven unseen benchmark datasets with manipulations distinct from those used in training, MuRF-FSL achieves a dataset-equal F1 of 0.9192, exceeding all evaluated reference-based and reference-free baselines while retaining seconds-level inference. The ablations identify SR as the largest contributor to localization accuracy, with smaller, combination-dependent refinements from the auxiliary branches. These results support the central premise: for every feature, the pattern shared by a strict majority of sources is the reference one never had, and no single source needs to be entirely clean for it to serve as the internal comparison reference.

AI USE STATEMENT

In this work, we used generative AI tools (large language models) to assist with translation, implementation and code, and conceptual and methodological feedback. We did not use generative AI tools to generate synthetic datasets, to formulate mathematical claims or assist in the writing of proofs, to propose or refine hypotheses, to design experiments, or to interpret results. Additionally, we used a generative AI tool to draft and edit parts of the research paper to improve readability, to search for and identify relevant literature, and to check reference formatting and citation consistency.

Parts of the manuscript were written first in Chinese by the authors and subsequently translated into English with the assistance of a generative AI tool. The authors reviewed and revised every translated passage and verified that it matches the intended technical content. We used a generative AI tool to help write and debug parts of the code implementing our method, the training pipeline, and the evaluation scripts. Every AI-assisted artifact was executed, tested, and verified by the authors, and all reported numbers were reproduced from the final code. We used a generative AI tool as a discussion partner when structuring the related-work argument, when naming and organising the symmetry properties of our method, and when checking terminology consistency between the main text and the appendix. The research problem, the task formalization, the identifiability assumptions, the architecture, the training procedure, and the evaluation protocol were designed and decided by the authors.

We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Fatemah Alqallaf, Stefan Van Aelst, Victor J Yohai, and Ruben H Zamar. Propagation of outliers in multivariate data. *The Annals of Statistics*, pp. 311–331, 2009.

Said Bahassine, Abdellah Madani, Mohammed Al-Sarem, and Mohamed Kissi. Feature selection using an improved chi-square for arabic text classification. *Journal of King Saud University Computer and Information Sciences*, 32(2):225–231, 2020.

Kimberly A. Barchard and Larry A. Pace. Preventing human error: The impact of data entry methods on data accuracy and statistical results. *Computers in Human Behavior*, 27(5):1834–1839, 2011. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2011.04.004. URL https://www.sciencedirect.com/science/article/pii/S0747563211000707. 2009 Fifth International Conference on Intelligent Computing.

Míriam Barrabés, Daniel Mas Montserrat, Margarita Geleta, Xavier Giró-i Nieto, and Alexander Ioannidis. Adversarial learning for feature shift detection and correction. *Advances in Neural Information Processing Systems*, 36:57597–57638, 2023.

Míriam Barrabés, Daniel Mas Montserrat, Kapal Dev, and Alexander G Ioannidis. Feature shift localization network. *arXiv preprint arXiv:2506.09101*, 2025.

Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from training data. *Advances in neural information processing systems*, 33:17605–17616, 2020.

Catherine L Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998.

Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

Luis M Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81–97, 2017.

Zehra Cataltepe, Chao Yuan, Claus Neubauer, Meltem Demirkus, and Hans-Gerd Brummel. A robust method to identify faults in correlated sensors in machine condition monitoring. In *2005 13th European Signal Processing Conference*, pp. 1–4. IEEE, 2005.

Qi Cheng, Pramod K Varshney, and Manoj K Arora. Logistic regression for feature selection and soft classification of remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 3(4): 491–494, 2006.

Tao Dai, Hongpu Hu, Yanli Wan, Quan Chen, and Yan Wang. A data quality management and control framework and model for health decision support. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1792–1796, 2015. doi: 10.1109/FSKD.2015.7382218.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

Nadir Omer Fadl Elssied, Othman Ibrahim, and Ahmed Hamza Osman. A novel feature selection based on one-way anova f-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, 7(3):625–638, 2014.

François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research*, 5(Nov):1531–1555, 2004.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

Konstantin Hopf and Sascha Reifenrath. Filter methods for feature selection in supervised machine learning applications–review and benchmark. *arXiv preprint arXiv:2111.12140*, 2021.

Xiaojuan Huang, Li Zhang, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Feature clustering based support vector machine recursive feature elimination for gene selection. *Applied Intelligence*, 48 (3):594–607, 2018.

Ramon Huerta, Thiago Mosqueiro, Jordi Fonollosa, Nikolai F Rulkov, and Irene Rodriguez-Lujan. Online decorrelation of humidity and temperature in chemical sensors for continuous monitoring. *Chemometrics and Intelligent Laboratory Systems*, 157:169–176, 2016.

Sean Kulinski, Saurabh Bagchi, and David I Inouye. Feature shift detection: Localizing which features have shifted via conditional distribution tests. *Advances in neural information processing systems*, 33:19523–19533, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

David D. Lewis. Reuters-21578 Text Categorization Test Collection, Distribution 1.0. Dataset, 1997. URL `https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter*, 17(2):1–16, 2016.

Lek-Heng Lim and Bradley J Nelson. What is an equivariant neural network. *arXiv preprint arXiv:2205.07362*, 2022.

Su Bin Lim, Swee Jin Tan, Wan-Teck Lim, and Chwee Teck Lim. A merged lung cancer transcriptome dataset for clinical predictive modeling. *Scientific data*, 5(1):180136, 2018.

Anjin Liu, Jie Lu, and Guangquan Zhang. Diverse instance-weighting ensemble based on region drift disagreement for concept drift adaptation. *IEEE transactions on neural networks and learning systems*, 32(1):293–307, 2020.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM transactions on knowledge discovery from data (TKDD)*, 6(1):1–39, 2012.

Chun Lo, Yechao Bai, Mingyan Liu, and Jerome P Lynch. Efficient sensor fault detection using group testing. *arXiv preprint arXiv:1501.04152*, 2015.

Viktor Losing, Barbara Hammer, and Heiko Wersing. Knn classifier with self adjusting memory for heterogeneous concept drift. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 291–300. IEEE, 2016.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

Ning Lu, Jie Lu, Guangquan Zhang, and Ramon Lopez De Mantaras. A concept drift-tolerant case-base editing technique. *Artificial Intelligence*, 230:108–133, 2016.

Sebastián Maldonado and Richard Weber. A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208–2217, 2009.

Sonia Moreno-Grau, Manvi Vernekar, Arturo Lopez-Pineda, Daniel Mas-Montserrat, Míriam Barrabés, Consuelo D Quinto-Cortés, Babak Moatamed, Ming Ta Michael Lee, Zhenning Yu, Kensuke Numakura, et al. Polygenic risk score portability for common diseases across genetically diverse populations. *Human Genomics*, 18(1):93, 2024.

Anam Mustaqeem, Syed Muhammad Anwar, Muhammad Majid, and Abdul Rashid Khan. Wrapper method for feature selection to classify cardiac arrhythmia. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3656–3659. IEEE, 2017.

Inzamam Mashood Nasir, Muhammad Attique Khan, Mussarat Yasmin, Jamal Hussain Shah, Marcin Gabryel, Rafał Scherer, and Robertas Damaševičius. Pearson correlation-based feature selection for document classification using balanced training. *Sensors*, 20(23):6793, 2020.

Jing Pan, Vincent Pham, Mohan Dorairaj, Huigang Chen, and Jeong-Yoon Lee. Adversarial validation approach to concept drift problem in user targeting automation systems at uber. *arXiv preprint arXiv:2004.03045*, 2020.

Maria Perera, Daniel Mas Montserrat, Miriam Barrabes, Margarita Geleta, Xavier Giro-i Nieto, and Alexander G Ioannidis. Generative moment matching networks for genotype simulation. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1379–1383. IEEE, 2022.

Jinchuan Qian, Zhihuan Song, Yuan Yao, Zheren Zhu, and Xinmin Zhang. A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 231:104711, 2022. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2022.104711. URL https://www.sciencedirect.com/science/article/pii/S0169743922002222.

Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Maria L Rizzo and Gábor J Székely. DISCO analysis: A nonparametric extension of analysis of variance. *The Annals of Applied Statistics*, 4(2):1034–1055, 2010. doi: 10.1214/09-AOAS245.

Peter J Rousseeuw and Wannes Van Den Bossche. Detecting deviating data cells. *Technometrics*, 60(2):135–145, 2018.

Ajanta Saha, Sotoudeh Sedaghat, Sarath Gopalakrishnan, Jose Waimin, Aiganym Yermembetova, Nicholas Glassmaker, Charilaos Mousoulis, Ali Shakouri, Alexander Wei, Rahim Rahimi, et al. A new paradigm of reliable sensing with field-deployed electrochemical sensors integrating data redundancy and source credibility. *Scientific Reports*, 13(1):3101, 2023.

Chris L Stork and Bruce R Kowalski. Distinguishing between process upsets and sensor malfunctions using sensor redundancy. *Chemometrics and intelligent laboratory systems*, 46(2):117–131, 1999.

Emma VA Sylvester, Paul Bentzen, Ian R Bradbury, Marie Clément, Jon Pearce, John Horne, and Robert G Beiko. Applications of random forest feature selection for fine-scale genetic population assignment. *Evolutionary applications*, 11(2):153–165, 2018.

Binh Tran, Mengjie Zhang, and Bing Xue. A pso based hybrid feature selection algorithm for high-dimensional classification. In *2016 IEEE congress on evolutionary computation (CEC)*, pp. 3801–3808. IEEE, 2016.

Jan N Van Rijn, Bernd Bischl, Luis Torgo, Bo Gao, Venkatesh Umaashankar, Simon Fischer, Patrick Winter, Bernd Wiswedel, Michael R Berthold, and Joaquin Vanschoren. Openml: A collaborative science platform. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 645–649. Springer, 2013.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Xiaoxin Yin, Jiawei Han, and Philip S Yu. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1048–1052, 2007.

Shujian Yu, Xiaoyang Wang, and José C Príncipe. Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels. *arXiv preprint arXiv:1806.10131*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

A Statistical Descriptor Network

MuRF-FSL adopts the pretrained Statistical Descriptor Network of FSL-Net (Barrabés et al., 2025) to transform empirical distributions with variable sample sizes into fixed-dimensional representations while preserving feature correspondence. It first pools samples from all sources to estimate shared per-feature normalization parameters and then separates the normalized samples back into their respective sources. The shared network ϕ_θ processes each source independently and produces three per-feature mappings: an explicit statistical mapping $\boldsymbol{H}_S^{(k)}$ comprising the mean, second- and third-order moments, median, standard deviation, mean absolute deviation, marginal histogram, and empirical cumulative distribution function. A moment extraction mapping $\boldsymbol{H}_M^{(k)}$ captures higher-order and cross-feature statistics through shallow convolutions and sample-mean pooling. A neural embedding mapping $\boldsymbol{H}_N^{(k)}$ learns richer distributional representations through deep residual convolutions and sample-mean pooling. These mappings are concatenated along the descriptor channel to obtain $\boldsymbol{H}^{(k)} = \phi_\theta\big(\mathcal{X}^{(k)}\big) = \Big[\boldsymbol{H}_S^{(k)}; \boldsymbol{H}_M^{(k)}; \boldsymbol{H}_N^{(k)}\Big] \in \mathbb{R}^{d\times B}$, after which all source descriptors are stacked as $\boldsymbol{H} = \operatorname{stack}_{k=1}^{K} \boldsymbol{H}^{(k)} \in \mathbb{R}^{K\times d\times B}$. This module captures the base information of each source, providing the basis for the subsequent construction of the reference.

B Auxiliary Enhancement Module

The auxiliary enhancement module groups CSM, CSN, and CSU after the SR construction in Section 3. It receives $\boldsymbol{H}$, $\boldsymbol{R}$, and the reference statistics $\boldsymbol{m}$, $\boldsymbol{s}$, $\overline{\boldsymbol{h}}$, and $\boldsymbol{\sigma}$, and returns $\boldsymbol{U}_{\text{aux}} \in \mathbb{R}^{K\times d\times D_a}$, where $D_a = D_m + D_n + D_u$. All three branches share parameters across sources and preserve the source and feature axes.

Cross-Source Statistical Moment Branch (CSM). To exploit the robust deviation, the average state of the remaining sources, and the cross-source dispersion simultaneously, we construct $\boldsymbol{a}_{\text{mom},j}^{(k)} = \big[\boldsymbol{R}_j^{(k)}; \operatorname{slog}(\overline{\boldsymbol{h}}_j^{(-k)}); \log(1+\boldsymbol{\sigma}_j^{(-k)})\big] \in \mathbb{R}^{3B}$. The three B-dimensional blocks describe complementary aspects of the cross-source context at position (k, j): $\boldsymbol{R}_j^{(k)}$ measures how far source k deviates from the robust consensus of its peers in standardized units, $\operatorname{slog}(\overline{\boldsymbol{h}}_j^{(-k)})$ measures the location of that consensus, and $\log(1+\boldsymbol{\sigma}_j^{(-k)})$ measures how dispersed the peers are, i.e., how reliable that consensus is. Because the peer mean can take either sign while the peer standard deviation is nonnegative, the former is compressed with the sign-preserving slog and the latter with $\log(1+\cdot)$. A shared predictor f_{mom} then maps this vector to a D_m-dimensional moment embedding $\boldsymbol{U}_{\text{mom},j}^{(k)} = f_{\text{mom}}\Big(\boldsymbol{a}_{\text{mom},j}^{(k)}\Big) = \phi\Big(\boldsymbol{W}_2\,\phi\Big(\operatorname{LN}\Big(\boldsymbol{W}_1\boldsymbol{a}_{\text{mom},j}^{(k)} + \boldsymbol{b}_1\Big)\Big) + \boldsymbol{b}_2\Big)$, where $\boldsymbol{W}_1, \boldsymbol{W}_2$ and $\boldsymbol{b}_1, \boldsymbol{b}_2$ are the weights and biases of two linear layers, LN denotes layer normalization over the hidden channels, and ϕ is a shared elementwise nonlinearity. The same f_{mom} is applied independently at every source-feature position, with the source and feature axes treated as batch axes and layer normalization computed within each position over the hidden channels. This branch therefore fuses only the descriptor channels at a fixed (k, j): it relates the current source to the peer statistics already summarized in $\boldsymbol{a}_{\text{mom},j}^{(k)}$, but it neither performs a further cross-source comparison nor exchanges information between neighboring features, the latter being handled by the cross-source neural branch.

Cross-Source Neural Branch (CSN). Cross-source statistics on a single feature are insufficient to describe shift patterns that co-occur across adjacent features. To this end, we concatenate the current source's base descriptor, robust residual, and leave-one-out robust center as $\boldsymbol{A}_{\text{n}}^{(k)} = \big[\operatorname{slog}(\boldsymbol{H}^{(k)}); \boldsymbol{R}^{(k)}; \operatorname{slog}(\boldsymbol{m}^{(-k)})\big] \in \mathbb{R}^{d\times 3B}$. All three components are compressed with the sign-preserving slog so that their magnitudes become comparable before the convolutions, and the robust center is included because $\boldsymbol{R}^{(k)}$ can only be interpreted relative to the reference against which it was formed. A shared network f_{n} then maps this tensor to $\boldsymbol{U}_{\text{n}}^{(k)} = f_{\text{n}}(\boldsymbol{A}_{\text{n}}^{(k)}) \in \mathbb{R}^{d\times D_n}$ by treating the $3B$ descriptor channels as input channels, the feature axis as the sequence axis, and the source axis as the batch axis, so that every source is processed independently with the same parameters. The tensor is first projected by a pointwise convolution, group normal-

ization, and a nonlinearity, yielding $\boldsymbol{Y}_0^{(k)} = \phi\Big(\mathrm{GN}\Big(\mathrm{Conv}_1\Big(\boldsymbol{A}_{\mathrm{n}}^{(k)\top}\Big)\Big)\Big)$, which fuses the descriptor channels at each feature position while leaving neighbouring features untouched. A stack of L_n shared residual blocks then models the feature axis. The ℓ-th block is $\boldsymbol{Y}_\ell^{(k)} = \phi\Big(\mathrm{GN}_2\Big(\mathrm{Conv}_\kappa\Big(\phi\Big(\mathrm{GN}_1\Big(\mathrm{Conv}_\kappa\Big(\boldsymbol{Y}_{\ell-1}^{(k)}\Big)\Big)\Big)\Big)\Big) + \boldsymbol{Y}_{\ell-1}^{(k)}\Big)$, where Conv_κ is a one-dimensional convolution with kernel size κ and length-preserving padding, and ϕ is a shared elementwise nonlinearity. The padding keeps the number of feature positions unchanged, so every input feature always corresponds to exactly one output position and the channel width stays equal to D_n throughout the stack, while the identity path forwards the incoming representation directly to each block output. Because these convolutions slide only along the feature axis, the branch exchanges information between neighbouring features but never between sources. Its receptive field is finite and grows with the depth of the stack, so each output position depends only on a local neighbourhood of the corresponding input feature. In contrast to the moment branch, which acts at fixed source and feature positions, this branch can therefore represent shift patterns that co-occur across adjacent features.

Consensus Stability and Uncertainty Branch (CSU). The reliability of the leave-one-out reference varies across sources and features. Let $\langle\cdot\rangle_B$ denote averaging over the B descriptor channels, and let $\mathrm{Std}_{B,\varepsilon}$ denote the channel-wise standard deviation with a numerical stabilization term. For feature j of source k, we construct the four-dimensional stability vector $\boldsymbol{u}_j^{(k)} = \big[\langle\log(1 + \boldsymbol{s}_j^{(-k)})\rangle_B;\ \langle|\boldsymbol{R}_j^{(k)}|\rangle_B;\ \mathrm{Std}_{B,\varepsilon}(\boldsymbol{R}_j^{(k)});\ \langle\log(1 + \frac{|\bar{\boldsymbol{h}}_j^{(-k)} - \boldsymbol{m}_j^{(-k)}|}{\boldsymbol{s}_j^{(-k)}})\rangle_B\big]$. The four components successively characterize the scale level of the leave-one-out reference, the strength of the current residual, the dispersion of the residual across descriptor channels, and the central inconsistency between the mean and the median. This vector is mapped by a linear layer, layer normalization, and a nonlinear activation to $\boldsymbol{U}_{\mathrm{unc},j}^{(k)} = f_{\mathrm{unc}}(\boldsymbol{u}_j^{(k)}) \in \mathbb{R}^{D_u}$, providing the prediction network with compact consensus-reliability information.

C Training and Validation Data Preparation and Manipulations

C.1 OpenML Datasets

We obtain heterogeneous tabular datasets from OpenML (Van Rijn et al., 2013). Only datasets with active status, at least 12,000 instances, and between 10 and 30,000 raw features are considered. Datasets overlapping with any held-out test benchmark are excluded before downloading. Target columns are removed, categorical attributes are numerically encoded while their feature-type metadata are retained, and constant features are discarded.

Missing values are handled using two candidate preprocessing strategies. Strategy A removes features with more than 40% missing values, whereas strategy B uses a 70% threshold. Both subsequently remove samples containing any remaining missing value. Only outputs retaining at least 12,000 samples and 10 nonconstant features are accepted. When both strategies are valid, one is selected randomly using the fixed data-preparation seed. Matrices containing more than 40 million cell values are partitioned into ten row-disjoint matrices for tractable loading and sampling.

C.2 Algebraic Simulated Datasets

The algebraic training and validation pools together contain 261 datasets, each with 12,000 samples and 1,000 continuous features. The generator balances three function families: third- or fourth-order polynomials, cosine functions, and logarithmic functions. For every feature, inputs are sampled independently as $x \sim \mathcal{U}(-5, 5)$ and function parameters remain fixed across samples. Polynomial features use $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, where the degree is sampled from $\{3, 4\}$, $a_4 = 0$ for degree three, and all active coefficients are sampled from $\mathcal{U}(-50, 50)$. Cosine features use $f(x) = a\cos(bx + c)$ with $a \sim \mathcal{U}(-50, 50)$ and $b, c \sim \mathcal{U}(-\pi, \pi)$. Logarithmic features use $f(x) = \log_a(x + 6)$ with $a \sim \mathcal{U}(2, 10)$.

C.3 Probabilistic Simulated Datasets

The probabilistic training and validation pools likewise contain 261 datasets of the same $12{,}000 \times 1{,}000$ shape. Each dataset is sampled from a Gaussian, Bernoulli, or Beta mixture with a number of components drawn uniformly from 1 to 100 and mixture weights drawn from a symmetric Dirichlet distribution. For Gaussian mixtures, component means are sampled from $\mathcal{N}(0, 1)$ and feature-wise variances from $\mathcal{U}(0.1, 1.1)$. For Bernoulli mixtures, success probabilities are sampled from $\mathcal{U}(0, 1)$. For Beta mixtures, both shape parameters are sampled from $\mathcal{U}(1, 2)$.

With probability 0.25, a probabilistic dataset undergoes between one and five additional transformation steps. Each step first applies either standardization or min–max normalization, multiplies the matrix by a random square matrix whose entries follow a Uniform, Gaussian, Beta, or Bernoulli distribution, and then applies ReLU, GELU, sigmoid, Tanh, or a signed logarithm. With probability 0.5, the transformed result is combined through a residual connection with either the preceding state or the initial matrix, optionally after normalizing that residual source. These transformations increase the diversity of marginal shapes and cross-feature dependencies represented in the training pool.

C.4 Training and Validation Split

Using seed 42, we form a training pool of $2{,}610$ matrices: $2{,}114$ OpenML, 248 algebraic, and 248 probabilistic. A disjoint validation pool contains 129 matrices: 103 OpenML, 13 algebraic, and 13 probabilistic. The held-out test datasets and testing manipulations are excluded from both pools. Every stored matrix is initially unshifted. Multi-source episodes, shifts, and supervision labels are generated online.

For each training episode, a base matrix is selected from the training pool, its samples and features are independently shuffled, and between $5{,}000$ and $20{,}000$ samples and between 8 and 256 features are drawn without replacement. After feature-wise normalization to $[0, 1]$, the number of sources K is sampled uniformly from $\{3, \ldots, 10\}$ and the samples are divided into $K + 1$ disjoint subsets, each containing at least 450 samples. The first K subsets form the observable sources, while the remaining subset is an auxiliary pool used only to generate shifts. For every source, the shift fraction is sampled from $\{0.05, 0.10, 0.25\}$ and shifted features are allocated subject to the per-feature strict-majority-normal constraint. Continuous features use T1–T7, categorical features use T3, T5, T6, and T8, and operation types are sampled uniformly. The validation pipeline is identical except that it draws from the validation pool and permits up to 512 features. The complete definitions of T1–T8 are provided in Table 3.

C.5 Training and Validation Manipulations

During training, we randomly select a base matrix from the training pool, randomly shuffle its sample order and feature order, draw without replacement between 5,000 and 20,000 samples and between 8 and 256 features, and then normalize each feature to $[0, 1]$. The number of sources K is sampled uniformly from 3 to 10, and the normalized samples are approximately evenly divided into $K + 1$ non-overlapping subsets, of which K subsets form the sources to be inspected and the remaining subset serves as an auxiliary sampling pool used only for simulating shifts. Each subset contains at least 450 samples. For each episode, we choose one shift ratio uniformly from $\{5\%, 10\%, 25\%\}$. This ratio determines the number of shifted features in every source, while feature sets and type-specific operations are selected subject to the per-feature strict-majority-normal constraint. The operation for continuous features is chosen uniformly from T1 to T7, and the operation for categorical features from T3, T5, T6, and T8. Target features of the same type within one source share a single operation, while different sources may use different operations. The validation procedure follows the same pipeline, but the base matrices come from the validation pool and the upper bound on the number of features is extended to 512. Both training and validation use fixed uniform operation sampling and do not adjust the operation probabilities according to the validation F1 score. The multiple operations are used to simulate a broad range of marginal and dependence shifts. The auxiliary sampling pool is not passed into MuRF-FSL and does not constitute reference data at inference time, and the final test benchmark does not participate in the selection of the training and validation operations. The specific operation definitions are as follows.

Table 3: Manipulation types applied to continuous and/or categorical features during training and validation. Adapted from FSL-Net (Barrabés et al., 2025).

Type	Mapping	Description	Shift	Data
T1	$\beta_j^{(k)} x_{i,j}^{(k)}, \beta_j^{(k)} \sim \mathrm{Uniform}(0,1)$	Each value is multiplied by a random number between 0 and 1.	$p_j \neq q_j^{(k)}$	Cont.
T2	$\beta_j^{(k)}(1-x_{i,j}^{(k)}) + (1-\beta_j^{(k)})x_{i,j}^{(k)}$, $\beta_j^{(k)} \sim \mathrm{Uniform}(0,1)$	Each value is replaced by a linear combination of x and $1-x$.	$p_j \neq q_j^{(k)}$	Cont.
T3	$x_{i,j}^{(k)} \sim \widehat{p}_{\mathcal{A},j}$	Feature j of source k is replaced by sampling from the auxiliary empirical marginal $\widehat{p}_{\mathcal{A},j}$.	$p_j = q_j^{(k)}$, $p_{\mathcal{C}} \neq q_{\mathcal{C}}^{(k)}$, $q_{\mathcal{C}}^{(k)} = \prod_{j\in\mathcal{C}} q_j^{(k)}$	Both
T4	$\mathrm{clip}_{[0,1]}(x_{i,j}^{(k)} + \epsilon_{i,j}^{(k)})$, $\epsilon_{i,j}^{(k)} \sim \mathcal{N}(\mu_j^{(k)}, (\sigma_j^{(k)})^2)$	Add Gaussian noise with $\mu_j^{(k)} \sim \mathrm{Uniform}(-0.2, 0.2)$ and $\sigma_j^{(k)} \sim \mathrm{Uniform}(0.001, 0.5)$.	$p_j \neq q_j^{(k)}$, $\mathbb{E}[q_j^{(k)}] \approx \mathbb{E}[p_j] + \mu_j^{(k)}$	Cont.
T5	$\mathrm{CNN}_k(\boldsymbol{X}^{(k)})_{i,j}$	Forward through a CNN with min–max normalization or binarization.	$p_j \neq q_j^{(k)}$	Both
T6	$\boldsymbol{x}_{i,\mathcal{C}}^{(k)} \sim \widehat{p}_{\mathcal{A},\mathcal{C}}$	As in T3, but all features in $\mathcal{C}$ are sampled simultaneously from $\widehat{p}_{\mathcal{A},\mathcal{C}}$.	$p_j = q_j^{(k)}$, $p_{\mathcal{C}} = q_{\mathcal{C}}^{(k)}$, $p \neq q^{(k)}$	Both
T7	$\mathrm{KNN}_{\mathcal{A},j}(\boldsymbol{x}_{i,-j}^{(k)})$, $\kappa_j^{(k)} \in \{1,2,3,4,7,8,9\}$	Predict feature j with KNN regression.	–	Cont.
T8	$\mathrm{KNN}_{\mathcal{A},j}(\boldsymbol{x}_{i,-j}^{(k)})$, $\kappa_j^{(k)} \in \{1,2,3,4,7,8,9\}$	Predict feature j with KNN classification.	–	Cat.

Here, $\mathcal{A} = \{\boldsymbol{a}_r\}_{r=1}^{N_{\mathcal{A}}}$ is the simulation-only auxiliary pool, $\widehat{p}_{\mathcal{A},J}$ is its empirical distribution on feature set J, $\mathcal{C} = \mathcal{C}_{k,\tau}$ denotes the selected features of type τ in source k for the corresponding table row, and CNN_k denotes the T5 convolution and postprocessing transform. As in FSL-Net, the Shift column uses idealized distributional relations: T3 samples each selected feature independently from its auxiliary empirical marginal $\widehat{p}_{\mathcal{A},j}$, whereas T6 samples the selected features jointly from $\widehat{p}_{\mathcal{A},\mathcal{C}}$. With a finite auxiliary pool, these empirical distributions only approximate the corresponding clean distributions. For T7 and T8, the neighbor count $\kappa_j^{(k)}$ is sampled independently for each target feature from $\{1,2,3,4,7,8,9\}$, and neighbors are found in $\mathcal{A}$ using every feature except j. All targets are predicted from an immutable source copy saved at the start of the KNN operation, so earlier replacements cannot affect later targets. For T5, the input and two random Tanh convolution outputs are averaged. The kernel sizes are chosen from $\{3,5,7\}$, and the kernel weights and biases are sampled from $\mathcal{N}(0, 0.5^2)$ and $\mathcal{N}(0, 0.1^2)$, respectively. T6 acts on the continuous and categorical target subsets separately. After all operations are completed, each source is clipped to $[0,1]$. The Shift column follows the intended distributional effects in FSL-Net. In particular, the inequalities for T3 and T6 require relevant feature dependencies and need not hold for every dataset.

D Testing Manipulation

The testing manipulations follow the E1–E10 benchmark taxonomy of Barrabés et al. (2023) and differ from the T1–T8 operations used for training and validation. Continuous datasets use E1, E2, E3, E4.1–E4.3, E5, E7, E8, and E9, while categorical datasets use E2, E3, E6.1–E6.3, E7, E8, and E10. E1, E2, E4, E5, E6, and E7 primarily alter marginal distributions. E3 independently permutes each selected feature across samples, preserving its empirical marginal while potentially disrupting feature dependencies. E8 applies one shared row permutation to all selected features, preserving their internal joint distribution while potentially disrupting their dependence with the remaining features. E9 and E10 replace continuous and categorical targets, respectively, with predictions obtained from the independent clean source by KNN regression or classification.

Each dataset is first normalized feature-wise to $[0,1]$ and randomly split into K observable sources and one independent clean source for every $K \in \{3,\ldots,10\}$. For each shift fraction $\rho \in \{0.05, 0.10, 0.25\}$, each observable source receives $\max(1, \lfloor \rho d \rfloor)$ shifted features while every feature remains normal in a strict majority of sources. For each K and ρ, we generate as many episodes as there are valid manipulation types and independently sample one valid operation for each observable source in every episode. This procedure yields 240 episodes per continuous dataset and 192 episodes per categorical dataset. The clean source is never passed to MuRF-FSL and is used only to construct E9 and E10 and as privileged input to reference-based baselines.

Table 4: Manipulation types applied to continuous and/or categorical features during benchmarking. Adapted from Barrabés et al. (2023).

Type	Mapping	Description	Shift	Data
E1	$x_{i,j}^{(k)} \sim \mathrm{Uniform}(0,1)$	Each value is substituted by a random number between 0 and 1.	$p_j \neq q_j^{(k)}$	Cont.
E2	$1 - x_{i,j}^{(k)}$	Each value is negated.	$p_j \neq q_j^{(k)}$, $\mathbb{E}[q_j^{(k)}] = 1 - \mathbb{E}[p_j]$	Both
E3	$P_j^{(k)} \boldsymbol{x}_j^{(k)}$	$P_j^{(k)}$ is a random permutation matrix applied to feature j of source k.	$p_j = q_j^{(k)}$, $p_{\mathcal{C}} \neq q_{\mathcal{C}}^{(k)}$, $q_{\mathcal{C}}^{(k)} = \prod_{j \in \mathcal{C}} q_j^{(k)}$	Both
E4.1–E4.3	$\mathrm{clamp}_{0,1}(x_{i,j}^{(k)} + \alpha\sigma_{i,j}^{(k)})$, $\sigma_{i,j}^{(k)} \sim \mathrm{Rademacher}(0.5)$	Add constant noise with a random sign. $\alpha \in \{0.02, 0.05, 0.1\}$ for E4.1–E4.3, respectively.	$p_j \neq q_j^{(k)}$, $\mathbb{E}[p_j] \approx \mathbb{E}[q_j^{(k)}]$	Cont.
E5	$\mathrm{round}(x_{i,j}^{(k)})$	Values are binarized.	$p_j \neq q_j^{(k)}$	Cont.
E6.1–E6.3	$b_{i,j}^{(k)}(1 - x_{i,j}^{(k)}) + (1 - b_{i,j}^{(k)})x_{i,j}^{(k)}$, $b_{i,j}^{(k)} \sim \mathrm{Bernoulli}(\nu)$	Values are negated with probability $\nu \in \{0.2, 0.4, 0.6\}$ for E6.1–E6.3, respectively.	$p_j \neq q_j^{(k)}$, $\mathbb{E}[q_j^{(k)}] = \nu + (1 - 2\nu)\mathbb{E}[p_j]$	Cat.
E7	$\mathrm{MLP}(x_{i,j}^{(k)})$	Forward through an MLP with min–max normalization or binarization.	$p_j \neq q_j^{(k)}$	Both
E8	$P^{(k)} \boldsymbol{X}_{\mathcal{C}}^{(k)}$	$P^{(k)}$ is a random permutation matrix applied to all selected features simultaneously.	$p_j = q_j^{(k)}$, $p_{\mathcal{C}} = q_{\mathcal{C}}^{(k)}$, $p \neq q^{(k)}$	Both
E9	$\mathrm{KNN}_{\mathcal{R}}(\boldsymbol{x}_{i,-j}^{(k)})$, $\kappa_j^{(k)} \in \{5, 6\}$	Predict feature with KNN (Regressor).	–	Cont.
E10	$\mathrm{KNN}_{\mathcal{R}}(\boldsymbol{x}_{i,-j}^{(k)})$, $\kappa_j^{(k)} \in \{5, 6\}$	Predict feature with KNN (Classifier).	–	Cat.

The original benchmark does not specify the E7 network architecture or the exact E9/E10 neighbor count. We therefore use explicit reproducible implementations for these operations. E7 applies an independent random one-hidden-layer Tanh network of width 16 to each selected feature, min–max normalizes the output, and thresholds it at 0.5 for categorical data. For E9 and E10, the neighbor count $\kappa_j^{(k)}$ is sampled independently for each target feature from $\{5, 6\}$. Squared Euclidean distances are computed using every feature except the current prediction target, and all targets are replaced from an immutable copy of the source so that earlier replacements cannot affect later ones.

Here, p is the clean distribution, $q^{(k)}$ is the manipulated distribution of source k, $\mathcal{C} = \mathcal{C}^{(k)}$ is its selected feature set, and $\mathcal{R}$ is the independent clean source used only to construct E9 and E10 and as privileged input to reference-based baselines. For E6, ν denotes the per-value flip probability and is distinct from the feature shift fraction ρ. As in the original table, the Shift column states idealized distributional effects: E3 and E8 preserve the relevant empirical distributions within each source exactly, but equalities with p and the E3 factorization hold only approximately for finite samples. Inequalities involving dependencies also require those dependencies to exist.

E Benchmarking Methods

All methods are evaluated on the same normalized multi-source episodes and produce a binary decision for every observable source and feature. The methods differ in the information available at inference time. MuRF-FSL and FSL-Net w/o Ref receive only the K observable sources, whereas FSL-Net w/ Ref, DataFix, MI, and SelectKBest additionally receive the independent clean source. The latter four methods are therefore privileged reference-based baselines rather than deployable reference-free methods.

E.1 Reference-Based Localization Methods

FSL-Net w/ Ref. FSL-Net (Barrabés et al., 2025) is a pairwise feature shift localization network. For each observable source $\mathcal{X}^{(k)}$, the independent clean source is supplied as the reference input and $\mathcal{X}^{(k)}$ as the query input. The same pretrained network is used for all sources, producing one d-dimensional probability vector per source and requiring K pairwise forward passes per episode. Probabilities are binarized at 0.5.

DataFix. DataFix (Barrabés et al., 2023) uses the iterative DF-Locate procedure. At each iteration, a random-forest classifier is trained to discriminate samples from the reference and query sources. Its predicted class probabilities are used to estimate the total variation distance, while impurity-based feature importance scores identify the features most responsible for the discrepancy. The highest-ranked features are removed and the classifier is retrained until the estimated divergence falls below the stopping criterion or half of the features have been removed. A final knee-detection step selects the stopping point along the removal trajectory. We use a margin of 0.03, rather than the original default of 0.01, and retain the hard feature mask returned by DF-Locate. This choice follows from evaluating three candidate margins under the same protocol, yielding dataset-equal Global F1 scores of 0.8743, 0.8886, and 0.8928 for 0.01, 0.02, and 0.03, respectively. Because the margin affects only this reference-based baseline, we adopt the setting that is most favourable to it.

E.2 Reference-Free Pairwise Adaptation

FSL-Net w/o Ref. Because the original FSL-Net requires a reference/query pair, we construct a reference-free adaptation using only the observable sources. For target source k, every other source $\ell \neq k$ is used in turn as the reference, producing $K-1$ probability vectors. Their elementwise mean is thresholded at 0.5 to obtain the prediction for source k. Repeating this process for all sources requires $K(K-1)$ pairwise forward passes. This baseline introduces multi-source information only through output-level averaging and does not jointly construct a $K \times d$ prediction.

E.3 Statistical Filter Baselines

Mutual Information. For each observable source, its samples and the clean-reference samples are pooled and assigned a binary domain label. MI ranks features according to their statistical dependence with this label. We classify features with MI scores above 0.02 as shifted, following the tuned setting reported by Barrabés et al. (2023).

SelectKBest. SelectKBest uses the same binary domain-classification construction. For continuous datasets it applies the ANOVA-F test, while for categorical datasets it applies the Chi-square test. Features with p-values below 0.01 are classified as shifted. Because the benchmark implementation of MI and SelectKBest was not released, both methods are reconstructed from Table 4 and Appendix G.1 of Barrabés et al. (2023). The domain-labelled two-sample formulation is our own reading, since the original benchmark interface accepts only a reference and a query set and no task label.

F HYPERPARAMETER SETTINGS AND MODEL SELECTION

This appendix records the configuration used for the final MuRF-FSL training run and benchmark evaluation. It reports adopted values, not a hyperparameter search space. Dataset preparation and manipulation definitions appear in Appendices C and D, while the baseline constructions appear in Appendix E.

F.1 DESCRIPTOR AND NETWORK CONFIGURATION

The pretrained descriptor network is shared across sources. Table 5 records its settings and those of the source-relative residual. Table 6 gives the additional components in the full configuration, including the optional auxiliary enhancement module.

Table 5: Shared statistical descriptor and source-relative residual settings.

Component	Hyperparameter	Value
Descriptor	Initialization	Public FSL-Net pretrained checkpoint. All 48 tensors matched.
Descriptor	Statistical measures	Mean, standard deviation, median, mean absolute deviation, moments of orders 2 and 3, histogram, and empirical CDF.
Descriptor	Histogram/CDF bins	100, numerical stability constant 10^{-5}.
Descriptor	Moment extractor	64 channels, kernel size 75, dilation 1, ReLU, BatchNorm1d without running statistics, and sample-mean pooling.
Descriptor	Neural embedding	Five residual blocks, 64 channels, kernel size 5, dilation 1, Tanh, BatchNorm1d without running statistics, and sample-mean pooling, no attention.
Descriptor	Input normalization	Per-feature standardization using all sources jointly for the main branch, per-feature min–max normalization to $[0, 1]$ for histogram and CDF.
Descriptor	Output width	$B = 334$ per feature.
SR	Robust reference	Leave-one-source-out channelwise median and $1.4826\,\mathrm{MAD}$, scale floor $s_{\min} = 0.05$ and numerical stability constant 10^{-5}.
SR	Residual	Signed logarithm of the standardized residual, clipped to $[-6, 6]$, output width 334.

Table 6: Auxiliary enhancement and prediction-network settings in the full configuration.

Component	Hyperparameter	Value
CSM	Widths and layers	Input $3B = 1{,}002$, hidden 256, output 64, two linear layers, LayerNorm, and GELU.
CSN	Convolutional stack	Input $3B = 1{,}002$, hidden/output 128, kernel size 7, two residual blocks, GroupNorm with 8 groups, and GELU.
CSU	Reliability embedding	Four scalar inputs, output width 16, LayerNorm, and GELU.
Full fused descriptor	Output width	$334 + 334 + 64 + 128 + 16 = 876$.
Prediction network	Convolutional stack	Input width 876, hidden width 128, kernel size 5, seven residual blocks, GroupNorm with 8 groups, and GELU.
Prediction network	Output and sharing	One-channel convolution followed by sigmoid. The same head is shared across all sources.
Parameters	Counts	Descriptor 212,224. Enhancement 862,768. Prediction network 1,265,409. Total 2,340,401.

F.2 Optimization and Validation

The final run used $2{,}610$ training matrices and 129 disjoint validation matrices, with the family-wise counts given in Appendix C. It processed $1{,}000{,}000$ synthetic episodes with one episode per optimization step. Validation covered all 129 matrices every $2{,}500$ episodes. The checkpoint maximizing mean validation F1 across manipulation types was used for all benchmarks. The checkpoint maximizing validation F1 was also saved but not used for the reported tests. Early stopping was disabled. The run took approximately 40.7 hours on one NVIDIA RTX 5090 (32 GB) with 8 CPUs and 64 GB of system memory.

Table 7: Optimization, loss, and checkpoint-selection settings.

Component	Hyperparameter	Value
Optimizer	Algorithm and weight decay	AdamW, weight decay 10^{-6}.
Optimizer	Learning rates	3×10^{-4} for enhancement and prediction networks, 10^{-6} for the pretrained descriptor.
Optimizer	Schedule and clipping	Both learning rates multiplied by 0.9995 after each validation, global gradient ℓ_2 norm capped at 0.5.
Loss	Main and auxiliary terms	BCEWithLogitsLoss plus pairwise descriptor-distance loss with weight 10^{-4}.
Loss	Margins and pair weight	Clean–shifted margin 2.0, different-operation shifted-pair margin 1.0, relative weight 0.1.
Run	Scale and precision	$1{,}000{,}000$ episodes, one episode per step, seed 42, float32 without mixed precision.
Validation	Interval and pool	Every $2{,}500$ episodes over all 129 validation matrices, no early stopping.
Checkpoint	Benchmark selection	`best_mean_op_f1_model.pth`, selected by mean per-operation validation F1. `best_model.pth` was saved separately by validation F1.

F.3 Episode and Benchmark Settings

Table 8 collects numerical episode settings. The full T1–T8 and E1–E10 definitions remain in Appendices C and D.

Table 8: Training, validation, and benchmark episode settings.

Component	Hyperparameter	Value
Training pool	Matrix counts	2,114 OpenML, 248 algebraic, 248 probabilistic, total 2,610.
Validation pool	Matrix counts	103 OpenML, 13 algebraic, 13 probabilistic, total 129.
Training episodes	Sources and samples	$K \sim \mathrm{Uniform}\{3, \ldots, 10\}$, 5,000–20,000 sampled rows, divided into $K+1$ subsets of at least 450 rows.
Training episodes	Feature count and shift fraction	$d' \in [8, 256]$ for training and $d' \in [8, 512]$ for validation, $\rho \in \{0.05, 0.10, 0.25\}$, $\max(1, \lfloor \rho d' \rfloor)$ shifted features per source.
Training episodes	Operation sampling	Continuous T1–T7, categorical T3, T5, T6, T8, uniform sampling, with one operation per feature type and source and strict-majority-normal allocation.
T5	Random convolution	Two Tanh layers, kernel sizes from $\{3, 5, 7\}$, weights $\mathcal{N}(0, 0.5^2)$ and biases $\mathcal{N}(0, 0.1^2)$.
T7/T8	KNN	Per-target $\kappa_j \in \{1, 2, 3, 4, 7, 8, 9\}$, all other features used, GPU distance batch size 256.
Benchmark	Episode grid	Eleven datasets, $K \in \{3, \ldots, 10\}$, $\rho \in \{0.05, 0.10, 0.25\}$, 240 episodes per continuous dataset and 192 per categorical dataset.
Benchmark	Manipulation settings	E4 amplitudes $0.02/0.05/0.10$, E6 flip probabilities $0.2/0.4/0.6$, E7 hidden width 16, E9/E10 per-target neighbor count from $\{5, 6\}$ with GPU distance batch size 256.
Benchmark	Construction seeds	1 for real datasets, 0 for Cosine and Polynomial.
Benchmark	Selection and timing	Prediction threshold 0.5, serial MuRF-FSL episode inference on one RTX 5090, mean and maximum wall-clock time per episode.

F.4 Baseline Parameters

The information-access rules and decision procedures are specified in Appendix E. Table 9 records the numerical and implementation settings used in those procedures.

Table 9: Baseline implementation and decision hyperparameters.

Method	Hyperparameter	Value
FSL-Net variants	Checkpoint and threshold	Public frozen `fslnet.pth`, threshold 0.5, deterministic serial GPU inference.
FSL-Net w/ Ref	Reference and calls	One independent clean source, K pairwise forward passes per episode.
FSL-Net w/o Ref	References and calls	All $K-1$ peer sources, mean of their probabilities, $K(K-1)$ pairwise forward passes per episode.
DataFix	Classifier and validation	`RandomForestClassifier(random_state=0)` with other sklearn defaults, 5-fold cross-validation and balanced accuracy.
DataFix	Stopping and removal	Margin 0.03, maximum removed-feature fraction 0.5, percentage 0.1 and alpha 1, step and test size unset.
DataFix	Knee detection and CPU	`knee-balanced`, window length 2, polynomial order 4, sensitivity 5, `n_jobs=1` per case, with 64 CPU processes over cases.
MI	Estimator and threshold	Continuous kNN mutual information with 3 neighbors, categorical contingency-table MI, seed 0, score threshold 0.02, GPU feature blocks of 64.
SelectKBest	Test and decision	ANOVA-F for continuous features, Chi-square for Founders and Reuters21578, $p < 0.01$ on GPU.

G SM-ONLY ABLATION SCORE AND F1 COMPUTATION

The SM row of Table 2 is a reference-free adaptation of the threshold-based statistical-measure comparison in FSL-Net (Barrabés et al., 2025). It uses the pretrained, frozen statistical descriptor network with a fixed scoring rule and no trainable enhancement or prediction head, optimizer, or episode training. Only the 206 explicit statistical-measure channels are scored: six scalar statistics (mean, second- and third-order moments, median, standard deviation, and mean absolute deviation), 100 empirical-CDF values, and 100 histogram values. The other 128 learned descriptor channels are excluded.

Let $S_{k,j,c}$ be statistical-measure channel c of feature j in source k, where $c \in [206]$. For each source, the leave-one-source-out reference is the channelwise peer median $M_{j,c}^{(-k)} = \mathrm{median}_{\ell \neq k}\, S_{\ell,j,c}$. We calculate a normalized squared difference and reduce it over channels to a source-feature score:

$$D_{k,j,c} = \frac{\left(S_{k,j,c} - M_{j,c}^{(-k)}\right)^2}{\left(\sum_{u=1}^{d} (M_{u,c}^{(-k)})^2\right)^{1/2} + 10^{-5}}, \qquad a_{k,j} = \frac{1}{206} \sum_{c=1}^{206} D_{k,j,c}.$$

The denominator's Euclidean norm is taken across features separately for each descriptor channel. This produces a $K \times d$ score matrix without a prespecified clean source.

The reduction and decision threshold were selected once on the 129 validation datasets and then fixed for all test datasets. One T1–T8 episode was sampled per validation dataset, giving 55,405 source-feature labels. For each of the mean and maximum channel reductions, 200 thresholds spaced between the pooled scores' first and 99th percentiles were evaluated by F1. Mean reduction attained the higher validation F1 (0.5168 versus 0.4174 for maximum reduction). Its selected threshold was $t = 0.003197051472400301$.

At test time, each source-feature pair is classified as shifted if $a_{k,j} \geq t$. Within each dataset, true positives, false positives, and false negatives are pooled over all episodes, sources, and features, and $F_{1,b} = 2\mathrm{TP}_b/(2\mathrm{TP}_b + \mathrm{FP}_b + \mathrm{FN}_b)$ is computed for dataset b. The benchmark contains 2,544 episodes across the eleven datasets. Table 10 reports their pooled F1 scores. The ablation-table value is the dataset-equal mean $\frac{1}{11}\sum_{b=1}^{11} F_{1,b} = 0.5728055946$, rounded to 0.5728. Confusion counts are not pooled across datasets, and no threshold is tuned on test data.

Table 10: Dataset-level F1 for the SM-only ablation.

Dataset	F1
Gas	0.754061
Covid	0.749792
Energy	0.557864
Musk2	0.509954
Scene	0.505742
MNIST	0.527707
Polynomial	0.486461
Cosine	0.361725
Dilbert	0.677842
Reuters21578	0.688546
Founders	0.481169
Dataset-equal mean	**0.572806**