SCAST: SPECTRAL CONSTRAINT-AWARE SELECTIVE TRANSFER FOR CROSS-SYSTEM LOG ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-system log anomaly detection transfers knowledge across systems with different event vocabularies and interaction structures, often under limited target-domain supervision. A key challenge is deciding which structural dependencies should be transferred before reducing source–target discrepancy. We present SCAST, a spectral constraint-aware selective transfer framework that performs dependency selection before alignment. SCAST maps heterogeneous log graphs into a shared semantic–structural space and decomposes their structural signals into multiscale spectral bands. Within each band, temporal direction and structural priors guide the modeling of dependency candidates, while a learned router separates features used for transfer from residual information. The selected band-wise representations are then fused into sample-level summaries for cross-domain alignment and anomaly scoring, rather than aligning an entangled whole-graph representation. We analyze the effects of cross-band fusion and the trade-off between reducing domain discrepancy and retaining predictive information.Experiments on four public log datasets show that SCAST performs competitively in cross-system transfer while remaining effective in single-system detection. Component and alignment-placement comparisons indicate that screening dependencies before alignment is more effective than aligning mixed representations. In an extended diagnostic model, frequency-resolved perturbations produce larger mean anomaly-logit changes for the screened branch than for residual and random perturbations in each of the three tested structural bands.

1 INTRODUCTION

System logs are widely used for monitoring, diagnosis, and anomaly detection in large-scale software systems. Recent deep models achieve strong single-system performance by learning sequential, semantic, pretrained, or graph-based log representations (Du et al., 2017; Meng et al., 2019; Almodovar et al., 2024; Guo et al., 2024). However, they commonly assume that training and test logs come from the same system, limiting their applicability when a detector must be transferred to a heterogeneous target system with scarce or unavailable anomaly labels.

Cross-system log anomaly detection is challenging because source and target systems may differ substantially in event vocabulary, template semantics, node scale, runtime environment, and structural interaction patterns. Existing methods reduce such discrepancies through domain adaptation, adversarial alignment, heterogeneous graph transfer, or temporal generative modeling (Han & Yuan, 2021; Wang et al., 2024; Gong et al., 2024; Zhang et al., 2026). Although effective, they typically align source and target representations before determining which structural dependencies are transferable. We refer to such indiscriminate alignment before dependency screening as *premature alignment*. It may leave anomaly-relevant dependencies entangled with domain-sensitive and potentially spurious associations, resulting in suboptimal transfer under large structural shifts. The key question is therefore not only how to align heterogeneous systems, but also which dependencies should be preserved, suppressed, and aligned.

Our key observation is that transferable and domain-sensitive dependencies may exhibit different behaviors across structural frequencies. Graph spectral components characterize structural patterns

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.

at different scales (Shuman et al., 2013; Zhang et al., 2023), and band-wise source–target shifts may point in opposing directions and become concealed after representation fusion. Motivated by this observation, we propose SCAST, a spectral constraint-aware selective transfer framework following a *decompose–screen–transfer* paradigm. SCAST first maps heterogeneous log graphs into a shared semantic-structural feature space and decomposes their structural signals into spectral bands. It then performs band-wise constraint-aware predictive dependency screening using temporal and structural priors, while routing complementary information into a residual branch. Finally, only the dynamically fused prior-screened dependency summaries are aligned across systems and used for anomaly scoring. SCAST thus replaces indiscriminate whole-representation alignment with frequency-resolved selective transfer.

We evaluate SCAST on HDFS, BGL, Thunderbird, and OpenStack under both single-system and cross-system settings. Experiments show that SCAST remains competitive in single-system detection and shows competitive performance across the reported transfer directions. The alignment-order comparison further demonstrates that cross-system alignment is beneficial and becomes more effective when applied to screened dependency summaries rather than mixed representations. Ablation, post-hoc perturbation, frequency-resolved routing, and efficiency analyses further support the proposed design.

The main contributions are summarized as follows:

- We introduce a frequency-resolved multiscale representation for heterogeneous log graphs, which decomposes structural signals before feature fusion. This reformulates direct cross-system structure alignment as band-wise dependency modeling, exposing scale-specific domain shifts and reducing structural entanglement caused by premature alignment.
- We propose a constraint-aware predictive dependency screening and selective transfer mechanism. Guided by temporal-direction and structural priors, SCAST screens predictive dependencies within each spectral band and routes complementary information into a residual branch, and aligns and scores only the dynamically fused screened summaries to support cross-system transfer.
- Extensive experiments on four public log datasets demonstrate the effectiveness of SCAST under both single-system and cross-system settings, with additional analyses supporting its key design choices.

2 Related Work

2.1 Cross-System Log Anomaly Detection

Existing cross-system log anomaly detection methods reduce source–target discrepancies through domain adaptation, adversarial alignment, and contrastive learning (Han & Yuan, 2021; Ganin et al., 2016; Tang & Guan, 2026). Although these methods promote domain-invariant representations, they may indiscriminately align transferable anomaly-related information with system-specific associations.

Recent approaches further model structural and temporal dependencies using heterogeneous graph transfer, time-aware adaptation, or generative temporal graphs (Wang et al., 2024; Gong et al., 2024; Zhang et al., 2026). However, they generally adapt node-level, graph-level, or temporally aggregated representations before assessing the transferability of individual dependencies. This leaves the transferability of frequency-specific dependencies underexplored.

2.2 Spectral Graph Representation Learning

Spectral graph learning provides a principled framework for characterizing graph signals at different scales of variation with respect to the underlying topology (Shuman et al., 2013). Within this framework, spectral graph wavelets apply scaled kernels to the graph Laplacian to construct localized, multiscale representations (Hammond et al., 2011). Subsequent work develops learnable spectral filters through polynomial parameterizations. ChebNet uses Chebyshev polynomials to implement localized graph convolutions (Defferrard et al., 2016), while BernNet employs Bernstein polynomial approximation to learn flexible spectral responses (He et al., 2021). Spectral analysis has also

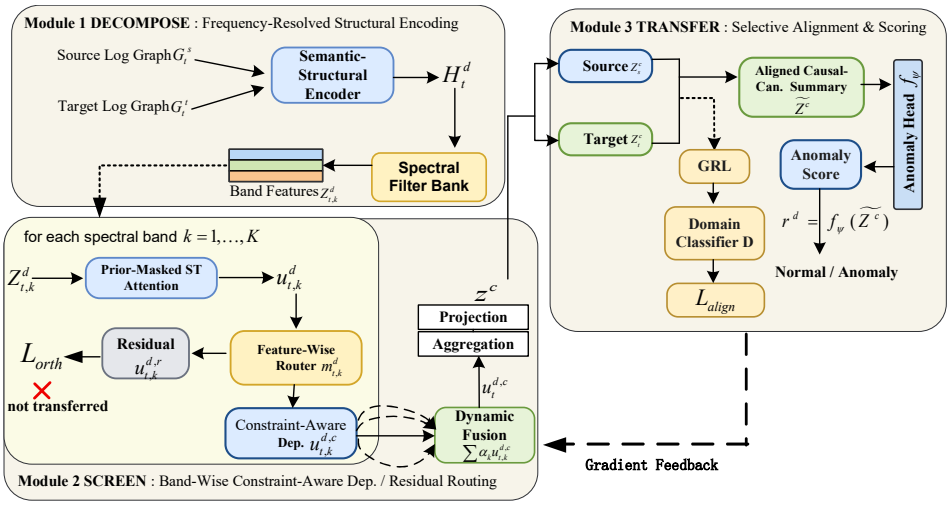

Figure 1: Overview of SCAST's *decompose–screen–transfer* pipeline: spectral decomposition, prior-guided dependency screening with residual separation, and summary alignment with dependency-sequence scoring.

been explored for invariant representation learning on dynamic graphs. SILD applies a Fourier transform to temporal sequences of node embeddings and learns spectral masks to capture invariant representations under distribution shifts (Zhang et al., 2023). These approaches operate on distinct notions of frequency: graph-Laplacian frequencies characterize signal variation across graph topology, whereas the Fourier frequencies used in SILD describe the temporal evolution of node representations.

3 METHOD

As illustrated in Figure 1, SCAST follows a *decompose–screen–transfer* paradigm. It first decomposes heterogeneous log-graph representations into multiple spectral bands. It then screens predictive dependencies within each band using temporal and structural priors. Finally, it aligns sample-level summaries of the screened dependencies and computes anomaly scores from the corresponding dependency sequences.

3.1 PROBLEM FORMULATION

We study cross-system log anomaly detection with scarce target-domain supervision. Let $\mathcal{D}_s$ and $\mathcal{D}_t$ denote the source and target domains. The two domains may differ in event vocabulary, template semantics, node scale, semantic distribution, and graph structure, making direct node-identity alignment infeasible. Therefore, SCAST performs transfer over dependency representations rather than raw template identities.

For each domain $d \in \{s, t\}$, a raw log stream is divided into sliding windows, and each sample is represented as a sequence of log graphs:

$$\mathcal{X}^d = \{G_1^d, G_2^d, \ldots, G_T^d\}, \qquad G_t^d = (\mathcal{V}_t^d, \mathcal{E}_t^d), \tag{1}$$

where nodes denote log templates and edges encode temporal co-occurrence or transition relations.

Let $\mathcal{N}_s$ and $\mathcal{N}_t$ denote the normal training sets in the source and target domains, respectively, and let $\mathcal{A}_t$ denote the optional set of sparsely labeled target anomalies. We assume that normal training windows are available in both domains, whereas target-domain anomaly labels are either unavailable or limited by the specified label budget. When available, $\mathcal{A}_t$ is used only for anomaly-score supervision and decision-boundary calibration under the same budget as the baselines.

The objective is to learn an anomaly-scoring function

$$f : \mathcal{X}^d \to [0, 1], \tag{2}$$

where a larger value indicates a higher anomaly risk.

3.2 DECOMPOSE: FREQUENCY-RESOLVED STRUCTURAL REPRESENTATION

Directly aligning whole-graph representations may mix dependencies across different structural scales and obscure scale-dependent cross-system shifts after feature fusion. SCAST therefore decomposes graph signals into spectral bands before constraint-aware predictive dependency screening and selective alignment.

For graph G_t^d, let $\mathbf{A}_t^d \in \mathbb{R}^{n_d \times n_d}$ and $\mathbf{X}_t^d \in \mathbb{R}^{n_d \times m}$ denote its adjacency and node-semantic matrices. A shared semantic-structural encoder Kipf & Welling (2017) produces

$$\mathbf{H}_t^d = \mathrm{GCN}\left(\mathbf{A}_t^d, \mathbf{X}_t^d\right), \tag{3}$$

placing different systems in a common feature space without requiring identical template sets or node identities.

Because systems may contain different numbers of templates, each domain uses a reference topology estimated only from its training snapshots. For N_d^{ref} reference graphs, we define

$$\mathbf{A}_{\mathrm{ref}}^d = \frac{1}{2N_d^{\mathrm{ref}}} \sum_{\ell=1}^{N_d^{\mathrm{ref}}} \left(\mathbf{A}_\ell^d + (\mathbf{A}_\ell^d)^\top\right), \mathbf{L}_{\mathrm{ref}}^d \quad = \mathbf{I} - (\mathbf{D}_{\mathrm{ref}}^d)^{-\frac{1}{2}} \mathbf{A}_{\mathrm{ref}}^d (\mathbf{D}_{\mathrm{ref}}^d)^{-\frac{1}{2}} = \mathbf{U}^d \mathbf{\Lambda}^d (\mathbf{U}^d)^\top, \tag{4}$$

where $\mathbf{D}_{\mathrm{ref}}^d$ is the corresponding degree matrix. The normalized Laplacians provide a common spectral range $[0, 2]$ for defining the filter bank, without requiring shared node identities or graph Fourier bases. This common frequency range does not imply that the same band has identical semantic or predictive roles across domains.

SCAST uses a Gaussian filter bank over the normalized Laplacian spectrum. For evenly spaced centers $c_k = 2(k-1)/(K-1)$, the response of band k is

$$h_k(\lambda) = \frac{\exp\left[-(\lambda - c_k)^2/(2\sigma^2)\right]}{\sum_{j=1}^K \exp[-(\lambda - c_j)^2/(2\sigma^2)]}, \qquad \sum_{k=1}^K h_k(\lambda) = 1. \tag{5}$$

For $K = 1$, we define $h_1(\lambda) = 1$, corresponding to the undecomposed single-band variant. The band-specific representation is then

$$\mathbf{Z}_{t,k}^d = h_k(\mathbf{\Lambda}^d)(\mathbf{U}^d)^\top \mathbf{H}_t^d, \qquad k = 1, \ldots, K, \tag{6}$$

where $h_k(\mathbf{\Lambda}^d)$ applies the band response to each eigenvalue, yielding band-weighted Fourier coefficients. Each domain's spectral basis is estimated from training data and fixed throughout training and inference. Attention and spatial masks operate in these fixed coordinates; the predictor is not generally invariant to eigenvector sign flips or rotations within repeated eigenspaces. Spatial-mask entries are therefore basis-dependent (Appendix F).

To illustrate the need for band-wise processing, let $\boldsymbol{\mu}_k^d = \mathbb{E}_{P_d}[\mathrm{Pool}(\mathbf{Z}_k^d)]$ and $\mathbf{\Delta}_k = \boldsymbol{\mu}_k^s - \boldsymbol{\mu}_k^t$. Premature fusion yields $\mathbf{\Delta}_{\mathrm{mix}} = \sum_k w_k \mathbf{\Delta}_k$. As illustrated in Fig. 2, these band-wise mean shifts may point in different directions. Consequently, a prematurely fused representation may exhibit a small aggregate mean shift even when individual band-wise shifts remain substantial. Spectral decomposition alone is therefore insufficient: predictive dependencies must be screened within each band before fusion and selective transfer. The geometric cancellation illustrated in Fig. 2 is a first-moment phenomenon. Its formal analysis is provided in Appendix A.2.

3.3 SCREEN: BAND-WISE CONSTRAINT-AWARE DEPENDENCY SELECTION AND RESIDUAL ROUTING

For each spectral band, SCAST learns spatial and temporal structural masks that encode structural preferences for dependency screening. Spatial indices denote spectral coordinates, not individual log-event identities. Let $\mathbf{\Theta}_k^d$ be the learnable spatial logits and $\boldsymbol{\rho}_k^d \in [0,1]^L$ the learnable strengths of the preceding L temporal lags. The masks are

$$\begin{aligned} \mathbf{M}_k^d &= \mathbf{I} + (\mathbf{1} - \mathbf{I}) \odot \sigma(\mathbf{\Theta}_k^d), \\ \mathbf{R}_{k,ij}^d &= \begin{cases} 1, & i = j, \\ \rho_{k,i-j}^d, & 1 \le i - j \le L, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7}$$

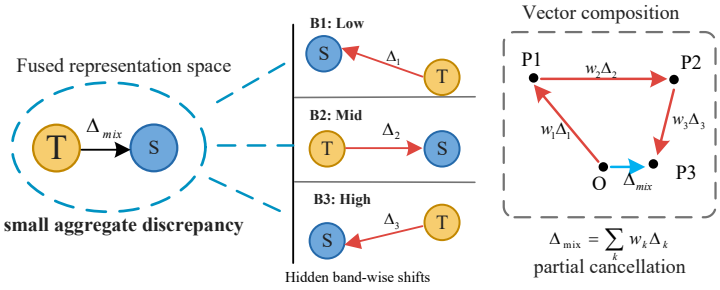

Figure 2: Hidden band-wise shift cancellation. Oppositely directed $\mathbf{\Delta}_k$ can yield a small aggregate $\mathbf{\Delta}_{\text{mix}}$ despite substantial band-specific discrepancies.

The spatial mask is initialized from the reference spectral support, whereas the lower-triangular temporal mask prevents future information leakage and imposes a temporal support restriction.

For the current graph, the spectral support is

$$\widehat{\mathbf{A}}_t^d = (\mathbf{U}^d)^\top \mathbf{A}_t^d \mathbf{U}^d. \tag{8}$$

Masked spatial and temporal attention then produces

$$\mathbf{V}_{t,k}^d = \mathrm{STAttn}_k\left(\mathbf{Z}_{1:t,k}^d, \widehat{\mathbf{A}}_{1:t}^d; \mathbf{M}_k^d, \mathbf{R}_k^d\right), \tag{9}$$

where spatial attention is modulated by $\mathbf{M}_k^d$ and the spectral support, while temporal attention is restricted by $\mathbf{R}_k^d$.

Because $\mathbf{V}_{t,k}^d$ may still contain domain-sensitive information, a feature-wise router decomposes it into a prior-screened dependency branch and a residual branch:

$$\mathbf{m}_{t,k}^d = \sigma\left(\mathrm{MLP}_m\left[\mathbf{Z}_{t,k}^d \| \mathbf{V}_{t,k}^d\right]\right), \mathbf{u}_{t,k}^{d,c} = \mathbf{m}_{t,k}^d \odot \mathbf{V}_{t,k}^d, \mathbf{u}_{t,k}^{d,r} = \left(\mathbf{1} - \mathbf{m}_{t,k}^d\right) \odot \mathbf{V}_{t,k}^d. \tag{10}$$

Here, c and r denote the prior-screened and residual branches, with $\mathbf{u}_{t,k}^{d,c} + \mathbf{u}_{t,k}^{d,r} = \mathbf{V}_{t,k}^d$. Only the router-selected c branch is used for alignment and anomaly scoring. An orthogonality regularizer penalizes squared cosine similarity between the pooled branch summaries.

Dynamic fusion is performed only after screening. With

$$e_{t,k}^d = a_k\left(\mathrm{Pool}(\mathbf{u}_{t,k}^{d,c})\right) + \beta_k, \alpha_{t,k}^{d,c} = \frac{\exp(e_{t,k}^d)}{\sum_{j=1}^K \exp(e_{t,j}^d)}, \mathbf{u}_t^{d,c} = \sum_{k=1}^K \alpha_{t,k}^{d,c} \mathbf{u}_{t,k}^{d,c}, \tag{11}$$

where $a_k(\cdot)$ is a learnable scoring function, dependencies from different structural scales are evaluated separately before transfer.

3.4 Transfer: Selective Alignment and Scoring of Screened Predictive Dependencies

SCAST aligns only the prior-screened dependency branch. Its sample-level summary and shared projection are

$$\mathbf{c}^{d,c} = \frac{1}{T}\sum_{t=1}^T \mathrm{Pool}(\mathbf{u}_t^{d,c}), \qquad \mathbf{z}^{d,c} = g_\phi(\mathbf{c}^{d,c}). \tag{12}$$

Adversarial alignment with a gradient reversal layer is defined as

$$\hat{y}_{\text{dom}}^d = D\big(\mathrm{GRL}(\mathbf{z}^{d,c})\big), \mathcal{L}_{\text{align}} = -\sum_{d\in\{s,t\}} \left[y_{\text{dom}}^d \log \hat{y}_{\text{dom}}^d + (1 - y_{\text{dom}}^d)\log(1 - \hat{y}_{\text{dom}}^d)\right]. \tag{13}$$

Applying this loss after screening restricts adversarial alignment to the selected summary. The residual summary is excluded from the alignment objective.

The anomaly score is also computed only from the prior-screened dependency sequence:

$$r^d = \sigma\left[f_\psi\left(\mathrm{TCN}\left(\{\mathrm{vec}(\mathbf{u}_t^{d,c})\}_{t=1}^T\right)\right)\right], \tag{14}$$

where $r^d \in [0,1]$ denotes anomaly confidence. The residual branch is excluded from both cross-domain alignment and anomaly scoring.

3.5 LEARNING OBJECTIVE

For normal samples from both domains, SCAST minimizes

$$\mathcal{L}_{\text{norm}} = -\sum_{x \in \mathcal{N}_s \cup \mathcal{N}_t} \log(1 - r(x)). \tag{15}$$

When sparse target-domain anomaly labels are available, SCAST combines anomaly-score supervision with a pairwise ranking term:

$$\mathcal{L}_{\text{anom}} = -\frac{1}{|\mathcal{A}_t|} \sum_{x_a \in \mathcal{A}_t} \log r(x_a) + \mathcal{L}_{\text{rank}}, \mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{N}_t||\mathcal{A}_t|} \sum_{x_n \in \mathcal{N}_t} \sum_{x_a \in \mathcal{A}_t} [\gamma - r(x_a) + r(x_n)]_+ . \tag{16}$$

where γ is the few-label ranking margin and $[z]_+ = \max(0, z)$.

The learned structural masks are regularized to favor parsimonious temporal dependencies and approximately acyclic spatial structures. Let

$$\overline{\mathbf{M}}_k^d = \mathbf{M}_k^d \odot (\mathbf{1} - \mathbf{I})$$

denote the off-diagonal spatial mask. The sparsity and acyclicity terms are

$$\begin{aligned} \mathcal{L}_{\text{sparse}} &= \sum_{d \in \{s,t\}} \frac{1}{2K} \sum_{k=1}^{K} \left(\frac{\|\overline{\mathbf{M}}_k^d\|_1}{n_d^2} + \frac{\|\boldsymbol{\rho}_k^d\|_1}{L} \right), \\ \mathcal{L}_{\text{dag}} &= \sum_{d \in \{s,t\}} \frac{1}{K} \sum_{k=1}^{K} \frac{\operatorname{tr}\left[\exp\left(\overline{\mathbf{M}}_k^d \odot \overline{\mathbf{M}}_k^d\right)\right] - n_d}{n_d^2}, \mathcal{L}_{\text{struct}} = \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}} + \lambda_{\text{dag}} \mathcal{L}_{\text{dag}}. \end{aligned} \tag{17}$$

The structural regularizer combines sparsity and approximate acyclicity constraints. The sparsity term discourages large off-diagonal mask weights and large lag weights, while the NOTEARS-style trace-exponential term Zheng et al. (2018) penalizes weighted cycles between spectral coordinates. The regularizer imposes sparsity and approximate acyclicity preferences on spectral-coordinate dependencies.

To regularize the geometric relationship between the two branches, we aggregate the residual representation as

$$\mathbf{c}^{d,r} = \frac{1}{T} \sum_{t=1}^{T} \text{Pool}\left(\sum_{k=1}^{K} \alpha_{t,k}^{d,c} \mathbf{u}_{t,k}^{d,r} \right). \tag{18}$$

The orthogonality regularizer is

$$\mathcal{L}_{\text{orth}} = \sum_{d \in \{s,t\}} \left[\frac{(\mathbf{c}^{d,c})^\top \mathbf{c}^{d,r}}{\|\mathbf{c}^{d,c}\|_2 \|\mathbf{c}^{d,r}\|_2} \right]^2. \tag{19}$$

For nonzero summaries, this loss encourages geometric orthogonality, not statistical independence or reduced mutual information. Branch complementarity follows from the router in Eq. 10, not this regularizer.

The complete objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{norm}} + \lambda_{\text{anom}} \mathcal{L}_{\text{anom}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{struct}} + \lambda_{\text{orth}} \mathcal{L}_{\text{orth}}. \tag{20}$$

Here, anomaly supervision encourages discrimination, $\mathcal{L}_{\text{struct}}$ imposes soft structural preferences, $\mathcal{L}_{\text{orth}}$ penalizes squared cosine similarity between the branch summaries, and $\mathcal{L}_{\text{align}}$ provides adversarial domain supervision on the selected summary. Together, these objectives train the router and the selected representation.

We analyze selective alignment at the representation level. Let U denote the mixed dependency representation before screening and $\widehat{C}$ the screened dependency summary used by SCAST. For each $X \in \{U, \widehat{C}\}$, let $\mathcal{H}_X$ denote the hypothesis class induced by representation X, and let $f_X \in \mathcal{H}_X$ be its predictor.

Under the standard binary domain-adaptation assumptions, the target risk associated with representation X satisfies the upper bound

$$\mathcal{B}(X, f_X) = R_s(f_X) + \frac{1}{2} d_{\mathcal{H}_X \Delta \mathcal{H}_X}\big(P_s(X), P_t(X)\big) + \lambda_X, \tag{21}$$

where $R_s(f_X)$ is the source-domain risk, $d_{\mathcal{H}_X \Delta \mathcal{H}_X}$ is the representation discrepancy, and

$$\lambda_X = \inf_{h \in \mathcal{H}_X} [R_s(h) + R_t(h)] \tag{22}$$

is the optimal joint error induced by representation X.

Define the source-risk change and joint-error change caused by screening as

$$\delta_s = R_s(f_{\widehat{C}}) - R_s(f_U), \qquad \delta_\lambda = \lambda_{\widehat{C}} - \lambda_U, \tag{23}$$

and define the discrepancy reduction as

$$\delta_{\text{disc}} = d_{\mathcal{H}_U \Delta \mathcal{H}_U}\big(P_s(U), P_t(U)\big) - d_{\mathcal{H}_{\widehat{C}} \Delta \mathcal{H}_{\widehat{C}}}\big(P_s(\widehat{C}), P_t(\widehat{C})\big). \tag{24}$$

If there exist nonnegative constants ϵ_s and ϵ_λ such that

$$\delta_s \le \epsilon_s, \qquad \delta_\lambda \le \epsilon_\lambda, \tag{25}$$

and

$$\delta_{\text{disc}} > 2(\epsilon_s + \epsilon_\lambda), \tag{26}$$

then

$$\mathcal{B}(\widehat{C}, f_{\widehat{C}}) < \mathcal{B}(U, f_U). \tag{27}$$

Therefore, selective alignment is beneficial when its reduction in cross-domain representation discrepancy exceeds the predictive information lost during screening.

4 Experiments

4.1 Datasets and Experimental Setup

Datasets. We evaluate SCAST on four public system-log datasets: HDFS (Xu et al., 2009; Zhu et al., 2023), BGL and Thunderbird (TB) (Oliner & Stearley, 2007; Zhu et al., 2023), and OpenStack (OS) (Zhu et al., 2023). They cover distributed storage, supercomputing, and cloud systems and differ in event vocabulary, template distribution, graph scale, runtime environment, and anomaly patterns.

We organize evaluation around transfer necessity, alignment placement, and predictive sensitivity. The fixed extension compares full SCAST, target-only learning, and pre-screen alignment on HDFS↔OpenStack using seeds 42, 43, and 44. Each target has 32 positive training windows, alongside 2,048 HDFS or 285 OpenStack normal training windows. Validation uses additional labels: 4,465 normal / 134 positive HDFS windows, or 95 normal / 33 positive OpenStack windows. These validation labels are part of the supervision budget.

The variants share target data, initialization rules, minibatch draws, and update budgets at each seed. Target-only removes source training and alignment; pre-screen alignment changes the input to the alignment branch while preserving the scoring path. Adam optimizes 15 epochs of 256 updates. Target validation PR-AUC selects the checkpoint, and validation F1 selects a logit threshold that is frozen for testing. Appendix B gives all preparation, training, metric, and artifact details. External baselines are not retrained.

Sensitivity, efficiency, graph-scale, and spectral-basis results are given in Appendices D, E, B.2, and F, respectively. To facilitate reproducibility, we provide the source code for the proposed method in the supplementary material.

4.2 Main Results

4.2.1 Single-System Results

Table 1(a) reports the original single-system F1 scores. SCAST obtains an average F1 of 0.961, with the largest displayed values on BGL and OpenStack. DLAD has the largest displayed values on HDFS and Thunderbird. The published baselines use heterogeneous encoders, data splits, label budgets, preprocessing, and threshold protocols.

Table 1: F1 scores for single-system detection and historical cross-system transfer. SCAST scores are from the original experiments. B/T/H denote BGL/Thunderbird/HDFS; OS is OpenStack; Avg. is the unweighted mean. These rows are descriptive context and are not the matched HDFS↔OpenStack extension.

(a) Single-system detection

Method	HDFS	BGL	TB	OS	Avg.
DeepLog	.915	.916	.743	.764	.835
LogAnomaly	.724	.821	.883	.835	.816
DLAD	**.979**	.957	**.983**	.896	.954
MTG-GAN	.947	.963	.963	.965	.960
SCAST	.928	**.964**	.969	**.984**	**.961**

(b) Historical cross-system transfer

Method	B→T	T→B	H→B	B→H	Avg.
LogTransfer	.906	.876	.846	.808	.859
SupCon	.926	.871	.794	.825	.854
LogTAD	.745	.700	.652	.719	.704
LogGT	.919	.798	.660	.962	.835
MTG-GAN	.937	.899	.831	.859	.882
LogDC	.940	.893	.821	.830	.871
SCAST	**.960**	**.931**	**.893**	**.973**	**.939**

Table 2: Historical BGL→TB component and two-by-two controls. P/R denote precision/recall; CC denotes constraint-aware screening. ΔF1 is the signed percentage-point change from full SCAST. The PR-AUC column is used in the two-by-two panel.

(a) Component ablation

Variant	P	R	F1	ΔF1
Full SCAST	.949	.971	.960	–
Pre-screen Align.	.909	.941	.925	-3.52
w/o Structural Reg.	.904	.809	.854	-10.63
w/o Domain Align.	.753	.990	.855	-10.44
w/o Dyn. Fusion	.907	.938	.922	-3.75
w/o Multi-band	.950	.904	.927	-3.33

(b) Two-by-two analysis

Variant	P	R	F1	PR-AUC
Base	.949	.804	.870	.944
Freq. only	.846	.971	.904	.843
CC only	.949	.895	.921	.942
Full SCAST	.949	.971	.960	.953

4.2.2 Cross-System Transfer Results

Table 1(b) reports the historical BGL→TB, TB→BGL, HDFS→BGL, and BGL→HDFS directions. The displayed SCAST F1 scores are 0.960, 0.931, 0.893, and 0.973, with an average of 0.939; the largest baseline average is MTG-GAN's 0.882.

We additionally evaluate HDFS→OpenStack and OpenStack→HDFS under the fixed protocol described above. Table 6 reports the aggregate results for Full SCAST, Pre-align, and Target-only variants. Full SCAST attains the highest mean F1, PR-AUC, and ROC-AUC in both directions. Compared with Target-only, the F1 and ROC-AUC gains are larger for HDFS→OpenStack, whereas the PR-AUC gain is slightly larger for OpenStack→HDFS (0.00617 vs. 0.00520).

4.3 Ablation Study

Table 2(a) retains the historical BGL→TB component controls. Removing structural regularization or domain alignment gives the largest reported F1 decreases. Pre-screen alignment gives F1 0.925, compared with 0.855 without domain alignment and 0.960 for full SCAST. These observations concern the original component study and do not establish a uniform transfer benefit across systems.

Table 2(b) reports the two-by-two analysis with CC denoting constraint-aware screening. Frequency decomposition and CC individually improve the displayed F1 over the base model; the combined model reaches F1 0.960 and recall 0.971. The full model also improves PR-AUC over the base model (0.953 versus 0.944), indicating consistent gains in this two-by-two comparison.

Transfer necessity and alignment placement. The target-only comparison measures the contribution of source training and alignment at a matched target supervision budget, while the pre-screen comparison tests whether presenting constrained representations to the discriminator improves pre-

diction. Appendix C, Table 6, reports the aggregate results for all three variants in the two transfer directions. F1 is evaluated at the frozen validation threshold; PR-AUC uses trapezoidal integration, and ROC-AUC is computed from the frozen test scores.

4.4 Frequency-Resolved Predictive Sensitivity Analysis

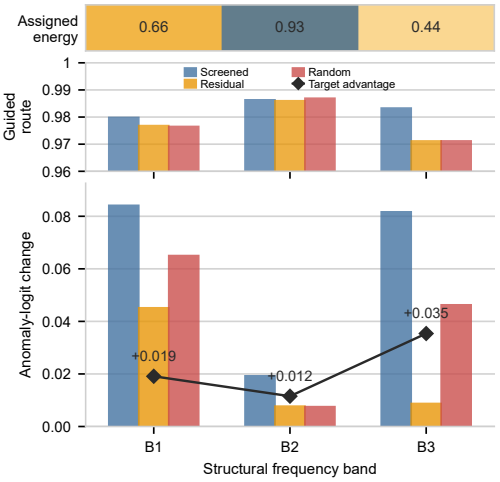

Figure 3: Frequency-resolved predictive sensitivity in an extended diagnostic model with a feature-wise gate and residual branch. Responses are measured under screened, residual, and random perturbations in three structural frequency bands.

Fig. 3 evaluates predictive sensitivity in an extended diagnostic model with an explicit feature-wise gate and residual branch. B1, B2, and B3 denote the low-, mid-, and high-frequency structural bands, respectively. The diagnostic compares screened and residual perturbations with energy-matched random perturbations within each designated band.
The panels report assigned perturbation energy, the guided-routing ratio, and absolute anomaly-logit changes. The guided-routing ratio measures the relative change in the screened summary, as defined in Appendix G. In the displayed results, screened perturbations produce larger mean absolute logit changes than residual and random perturbations in all three bands. For target-domain records, the screened response exceeds the larger of the two comparison responses by 0.019, 0.012, and 0.035 for B1–B3, respectively. These differences quantify sensitivity within the evaluated diagnostic configuration.

5 Conclusion

We presented SCAST, a spectral constraint-aware framework for cross-system log anomaly detection. SCAST decomposes heterogeneous log graphs into structural frequency bands, screens dependency candidates within each band, and aligns the resulting screened summaries before anomaly scoring. Experiments on four public log datasets show that this decompose–screen–transfer design improves cross-system transfer over target-only learning and pre-screen alignment while maintaining competitive single-system detection. Ablation studies support the roles of structural regularization, selective alignment, dynamic fusion, and multi-band representation.The extended diagnostic sensitivity analysis reports larger mean anomaly-logit changes for the screened branch than for residual and random perturbations in the tested bands.Taken together, the matched-control and ablation results support frequency-resolved selective transfer as an effective approach under the evaluated settings.

AI use statement

Generative AI tools were used to assist with literature review and LaTeX formatting. The authors independently verified all scientific content, experimental results, citations, and the final manuscript text, and retain full responsibility for the accuracy and integrity of the work.

References

Crispin Almodovar, Fariza Sabrina, Sarvnaz Karimi, and Salahuddin Azad. Logfit: Log anomaly detection using fine-tuned language models. *IEEE Transactions on Network and Service Management*, 21(2):1715–1723, 2024.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 1285–1298, 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Kun Gong, Senlin Luo, Limin Pan, Linghao Zhang, Yifei Zhang, and Haomiao Yu. Logeta: time-aware cross-system log-based anomaly detection with inter-class boundary optimization. *Future Generation Computer Systems*, 157:16–28, 2024.

Hongcheng Guo, Jian Yang, Jiaheng Liu, Jiaqi Bai, Boyang Wang, Zhoujun Li, Tieqiao Zheng, Bo Zhang, Junran Peng, and Qi Tian. Logformer: A pre-train and tuning pipeline for log anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 135–143, 2024.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and computational harmonic analysis*, 30(2):129–150, 2011.

Xiao Han and Shuhan Yuan. Unsupervised cross-system log anomaly detection via domain adaptation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3068–3072, 2021.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14239–14251, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/76f1cfd7754a6e4fc3281bcccb3d0902-Abstract.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Weibin Meng, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun, and Rong Zhou. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4739–4745. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/658. URL https://doi.org/10.24963/ijcai.2019/658.

Adam Oliner and Jon Stearley. What supercomputers say: A study of five system logs. In *37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07)*, pp. 575–584. IEEE, 2007.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

Pan Tang and Yepeng Guan. Cross-system log anomaly detection based on domain adversarial adaptation and batch contrastive learning. *Future Generation Computer Systems*, 175:108022, 2026. doi: 10.1016/j.future.2025.108022.

Peipeng Wang, Xiuguo Zhang, Zhiying Cao, Weigang Xu, and Wangwang Li. LogGT: cross-system log anomaly detection via heterogeneous graph feature and transfer learning. *Expert Systems with Applications*, 251:124082, 2024.

Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael I Jordan. Detecting large-scale system problems by mining console logs. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 117–132, 2009.

Jing Zhang, Chao Luo, and Rui Shao. MTG-GAN: A masked temporal graph generative adversarial network for cross-domain system log anomaly detection. *IEEE Transactions on Network and Service Management*, 2026. doi: 10.1109/TNSM.2026.3654642. URL https://doi.org/10.1109/TNSM.2026.3654642. Early Access.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. Spectral invariant learning for dynamic graphs under distribution shifts. *Advances in Neural Information Processing Systems*, 36:6619–6633, 2023.

Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9472–9483. Curran Associates, Inc., 2018.

Jieming Zhu, Shilin He, Pinjia He, Jinyang Liu, and Michael R Lyu. LogHub: A large collection of system log datasets for ai-driven log analytics. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 355–366. IEEE, 2023.

A Supplementary Theoretical Analysis

This appendix provides two complementary analyses of selective alignment in SCAST. Section A.1 derives the conditional target-risk comparison used in the main paper. Section A.2 gives a geometric analysis showing how fixed-weight fusion can conceal opposing band-wise mean shifts.

A.1 Formal Assumptions and Selective-Alignment Bound

We consider binary classification with source domain P_s and target domain P_t. For each representation $X \in \{U, \widehat{C}\}$, the representation map is fixed and induces a hypothesis class $\mathcal{H}_X$. The source and target domains may have different marginal distributions. Possible disagreement between their labeling functions is represented by the optimal joint error

$$\lambda_X = \inf_{h \in \mathcal{H}_X} \left[R_s(h) + R_t(h) \right], \tag{28}$$

where $R_d(h)$ denotes the expected classification risk of predictor h on domain $d \in \{s, t\}$.

Under the standard domain-adaptation assumptions, the representation-specific target-risk bound is

$$\mathcal{B}(X, f_X) = R_s(f_X) + \frac{1}{2} d_{\mathcal{H}_X \Delta \mathcal{H}_X}\big(P_s(X), P_t(X)\big) + \lambda_X. \tag{29}$$

Here, $d_{\mathcal{H}_X \Delta \mathcal{H}_X}$ denotes the symmetric- difference hypothesis-class discrepancy in the representation space induced by X. The two representations are:

- U, the mixed dependency representation before dependency screening; and
- $\widehat{C}$, the screened dependency summary used by SCAST for selective alignment and anomaly scoring.

We define

$$\delta_s = R_s(f_{\widehat{C}}) - R_s(f_U), \tag{30}$$

$$\delta_{\text{disc}} = d_{\mathcal{H}_U \Delta \mathcal{H}_U}\big(P_s(U), P_t(U)\big) - d_{\mathcal{H}_{\widehat{C}} \Delta \mathcal{H}_{\widehat{C}}}\big(P_s(\widehat{C}), P_t(\widehat{C})\big), \tag{31}$$

and

$$\delta_\lambda = \lambda_{\widehat{C}} - \lambda_U. \tag{32}$$

The quantity δ_s measures the source-risk cost of screening, δ_{disc} measures the reduction in cross-domain representation discrepancy, and δ_λ measures the change in the optimal joint error.

Proposition A.1 (Conditional benefit of selective alignment). Suppose that there exist nonnegative constants ϵ_s and ϵ_λ such that

$$\delta_s \le \epsilon_s, \qquad \delta_\lambda \le \epsilon_\lambda. \tag{33}$$

If

$$\delta_{\text{disc}} > 2(\epsilon_s + \epsilon_\lambda), \tag{34}$$

then

$$\mathcal{B}(\widehat{C}, f_{\widehat{C}}) < \mathcal{B}(U, f_U). \tag{35}$$

Proof. Using Eq. (29), we obtain

$$\begin{aligned} &\mathcal{B}(\widehat{C}, f_{\widehat{C}}) - \mathcal{B}(U, f_U) \\ &= R_s(f_{\widehat{C}}) - R_s(f_U) \\ &\quad + \frac{1}{2}\left[d_{\mathcal{H}_{\widehat{C}} \Delta \mathcal{H}_{\widehat{C}}}\big(P_s(\widehat{C}), P_t(\widehat{C})\big) - d_{\mathcal{H}_U \Delta \mathcal{H}_U}\big(P_s(U), P_t(U)\big) \right] \\ &\quad + \lambda_{\widehat{C}} - \lambda_U \\ &= \delta_s - \frac{1}{2}\delta_{\text{disc}} + \delta_\lambda. \end{aligned} \tag{36}$$

By Eq. (33) and Eq. (34),

$$\begin{aligned}\delta_s - \frac{1}{2}\delta_{\text{disc}} + \delta_\lambda &\le \epsilon_s - \frac{1}{2}\delta_{\text{disc}} + \epsilon_\lambda \\ &< \epsilon_s - (\epsilon_s + \epsilon_\lambda) + \epsilon_\lambda = 0.\end{aligned} \tag{37}$$

Therefore,

$$\mathcal{B}(\widehat{C}, f_{\widehat{C}}) - \mathcal{B}(U, f_U) < 0, \tag{38}$$

which proves Eq. (35). □

Interpretation. Proposition A.1 characterizes a sufficient trade-off condition. Screening may increase source risk and joint error, but selective alignment remains favorable when the reduction in representation discrepancy is larger than twice the sum of these costs.

The proposition compares representation-level upper bounds. It does not establish that the quantities δ_{disc} or λ_X are exactly known in practice, and it does not imply that an adversarial domain classifier reaches the optimal discrepancy reduction. In SCAST, the structural regularizer and residual routing are intended to limit the loss of predictive information, but this intention is not a theoretical guarantee.

A.2 Frequency-Resolved Mean-Shift Analysis

This section studies a fixed-weight idealization to explain why frequency-wise processing can be useful before cross-band fusion. The analysis concerns first-moment shifts only and does not directly characterize SCAST's sample-dependent dynamic routing.

For domain $d \in \{s, t\}$, let $\mathbf{H}_t^d \in \mathbb{R}^{n_d \times p}$ be the graph representation at time step t. Let

$$\mathbf{L}_{\text{ref}}^d = \mathbf{U}^d \mathbf{\Lambda}^d (\mathbf{U}^d)^\top \tag{39}$$

be the eigendecomposition of the reference graph Laplacian. The representation in spectral band k is

$$\mathbf{Z}_{t,k}^d = h_k(\mathbf{\Lambda}^d)(\mathbf{U}^d)^\top \mathbf{H}_t^d, \qquad k = 1, \dots, K. \tag{40}$$

Because the two systems may have different numbers of nodes, define the fixed-dimensional pooled summary

$$\mathbf{q}_{t,k}^d = \text{Pool}(\mathbf{Z}_{t,k}^d) \in \mathbb{R}^p. \tag{41}$$

Let $\mathbf{q}_k^d$ denote the random band summary induced by sampling a log-graph sequence from domain d. Define the band-wise mean shift

$$\mathbf{\Delta}_k = \mathbb{E}_{P_s}[\mathbf{q}_k^s] - \mathbb{E}_{P_t}[\mathbf{q}_k^t]. \tag{42}$$

Consider a hypothetical premature fusion with fixed, nonnegative, domain-independent weights:

$$\mathbf{q}_{\text{mix}}^d = \sum_{k=1}^K w_k \mathbf{q}_k^d, \qquad w_k \ge 0, \qquad \sum_{k=1}^K w_k = 1. \tag{43}$$

The resulting mean shift is

$$\mathbf{\Delta}_{\text{mix}} = \sum_{k=1}^K w_k \mathbf{\Delta}_k. \tag{44}$$

Proposition A.2 (Geometric cancellation under premature fusion). Under the fixed-weight setting above,

$$\begin{aligned}\|\mathbf{\Delta}_{\text{mix}}\|_2^2 = &\sum_{k=1}^K w_k^2 \|\mathbf{\Delta}_k\|_2^2 \\ &+ 2 \sum_{1 \le i < j \le K} w_i w_j \mathbf{\Delta}_i^\top \mathbf{\Delta}_j.\end{aligned} \tag{45}$$

If two positively weighted bands have $\mathbf{\Delta}_i^\top \mathbf{\Delta}_j < 0$, their cross-term is negative and reduces the squared aggregate mean shift.

Proof. The result follows by expanding $\| \sum_k w_k \mathbf{\Delta}_k \|_2^2$. If $\mathbf{\Delta}_i^\top \mathbf{\Delta}_j < 0$ and $w_i, w_j > 0$, then

$$2 w_i w_j \mathbf{\Delta}_i^\top \mathbf{\Delta}_j < 0.$$

Therefore, opposing band-wise shifts can partially cancel after premature fusion. □

Scope of the proposition. Proposition A.2 is a geometric statement about fixed-weight fusion and first-moment shifts. It does not imply small full-distribution discrepancy, lower target risk, or causal identification. SCAST uses sample-dependent post-screening weights, so this proposition serves as a motivation for inspecting and screening dependencies within each band before dynamic fusion; it is not a direct performance guarantee for the dynamic router.

B Datasets and Experimental Protocol

B.1 Datasets and Graph Construction

The experiments use four public system-log datasets: HDFS (Xu et al., 2009; Zhu et al., 2023), BGL and Thunderbird (TB) (Oliner & Stearley, 2007; Zhu et al., 2023), and OpenStack (OS) (Zhu et al., 2023), covering distributed storage, supercomputing, and cloud systems. Dataset sizes below refer to the available prepared partitions, rather than every record in the raw benchmarks.

BGL, TB, and OS snapshots are formed from rows of 20 event IDs. Consecutive IDs define symmetric binary edges; duplicate transitions are collapsed and self-loops are retained. HDFS uses supplied adjacency snapshots. Each model input contains $T = 10$ consecutive snapshots. The historical input/next-snapshot loader uses sliding windows and produces $S - T$ inputs from a file of S snapshots. The fixed HDFS↔OS evaluation instead selects nonoverlapping inputs under the split rules in Section B.3. These two window counts describe different sampling protocols and should not be interchanged.

B.2 Prepared Graph Scale and Sparsity

Table 3 gives the fixed template vocabulary size N and the complete prepared normal-training partition sizes. These partition-level counts are distinct from the selected training budgets in Table 5.

Table 3: Prepared normal-training partition sizes. N is the fixed template vocabulary size, S is the number of stored snapshots, and the window count follows the sliding input/next-snapshot loader with $T = 10$.

Dataset	N	Snapshots S	Input windows $S - T$
HDFS	29	401,921	401,911
BGL	157	56,470	56,460
TB	851	13,443	13,433
OS	56	3,766	3,756

An active node has at least one nonzero incident adjacency entry, including a self-loop; this count agrees with observed valid event IDs in the sequence-based partitions. Let E count distinct undirected non-self edges and ℓ count self-loops. Adjacency nonzeros equal $2E + \ell$, and off-diagonal density is $2E/[N(N-1)]$, normalized by the fixed vocabulary rather than the active-node count. A window union merges the node and edge sets across an input and is descriptive, not the graph passed to one model time step. Table 4 averages these quantities over the complete normal-training partitions and their eligible sliding-window unions.

Table 4: Mean graph statistics for the four prepared normal-training partitions. Density is scaled by 10^5. E excludes self-loops, which are reported separately as ℓ. A 10-snapshot union is descriptive and is not the graph passed to a single model time step.

Dataset	Unit	Active nodes	E	ℓ	Density $\times 10^5$
HDFS	Snapshot	7.26	6.52	3.87	1606.54
HDFS	10-snapshot union	9.78	15.76	5.84	3882.08
BGL	Snapshot	1.30	0.32	1.03	2.62
BGL	10-snapshot union	1.61	0.72	1.36	5.89
TB	Snapshot	8.42	9.55	1.88	2.64
TB	10-snapshot union	18.29	29.58	4.59	8.18
OS	Snapshot	9.93	10.91	1.09	708.15
OS	10-snapshot union	29.17	47.28	2.07	3069.96

OS has the largest mean active-node and edge counts, whereas HDFS has the highest vocabulary-normalized density. BGL and TB have similar snapshot densities despite different active graph sizes. Thus, a common temporal length does not imply comparable graph scale.

B.3 Fixed HDFS–OpenStack Splits and Supervision

Split construction. HDFS retains the supplied normal/anomalous adjacency partitions. Inputs are nonoverlapping within each file; training uses the selected subset in Table 5, while validation and test retain all eligible inputs. Raw block identifiers are unavailable, so upstream block independence cannot be certified from the adjacency arrays alone.

OS uses the available sequence table with source and log-line ranges, before the previous model-dependent filtering. Rows within each source stream are ordered by starting line and split 60/20/20 into training, validation, and test. Snapshots crossing a split boundary are purged, and successive evaluated inputs have disjoint raw-line ranges. An input is positive if any included snapshot has the local warning-derived anomaly label; these are proxy labels rather than independently verified fault annotations. Training retains all eligible normal inputs and samples the positive budget from 100 eligible positives.

Table 5: Selected training and complete validation/test window counts for the fixed HDFS–OS evaluation; each entry lists normal/anomalous windows.

Target	Training Normal/Anom.	Validation Normal/Anom.	Test Normal/Anom.
HDFS	2,048 / 32	4,465 / 134	11,164 / 336
OS	285 / 32	95 / 33	95 / 33

The counts apply when the listed domain is the target; transfer training uses only normal windows from the source domain. Positive training labels and labeled validation data are separate sources of supervision. The selected data and label budgets are fixed across seeds and internal controls.

Features, reference graphs, and indexing. Both domains use the supplied 28-dimensional template-feature matrices. Each reference topology is the mean adjacency over that domain's selected normal training windows. One-based OS event IDs are converted to zero-based matrix coordinates, and invalid indices raise an error; HDFS already uses adjacency tensors. The supplied features are preserved, although the available artifacts do not reconstruct every upstream semantic-encoding and raw-log preprocessing step.

B.4 Model, Optimization, and Evaluation

Architecture. Each domain uses two GCN layers of width 64, four spatial and four temporal attention heads per band, and a band-weight network with hidden width 64. The TCN has channels $(256, 256, 128)$ and kernel size 3; its output and scoring hidden width are 128. The shared projector has hidden/output widths 128/64, and the domain discriminator has hidden width 64. Dropout is 0.1. Reference bases and semantic matrices remain fixed while encoders and masks are optimized.

Optimization. Each run uses Adam (Kingma & Ba, 2015) at 2×10^{-4} for 15 epochs of 256 updates, with gradient-norm clipping at 1. An update samples eight normal and eight positive target windows with replacement; transfer variants additionally sample eight normal source windows. Role-specific random generators match target draws across variants at each seed. These are fixed update budgets, not full-data epochs. We use independent seeds 42, 43, and 44 and set

$$K = 3, \quad L = 3, \quad \sigma = 0.35, \quad \lambda_{\text{anom}} = 0.08, \quad \gamma = 0.08,$$
$$\lambda_{\text{align}} = 0.005, \quad \lambda_{\text{sparse}} = 2 \times 10^{-5}, \quad \lambda_{\text{dag}} = 10^{-6}.$$

The GRL multiplier is 0.1. Here L is the maximum lag of the within-band attention block, not the entire TCN scorer; it is distinct from the self-loop count ℓ in Section B.2.

Matched controls. Target-only preserves the target initialization, architecture, selected data, label budget, and update budget while removing the source encoder and alignment loss. Pre-screen alignment (Pre-align) retains both encoders and the full scoring path but sends unscreened band features to the alignment branch before pooling and projection. Its scoring heads remain separate; only the representation supplied to alignment changes. The three variants provide the internal comparison reported in Table 6.

Metrics and model selection. We report precision (P), recall (R), their harmonic mean F1, and areas under the ROC and precision–recall curves (ROC-AUC and PR-AUC) where available. PR-AUC uses trapezoidal integration; average precision is computed separately. The target validation set selects the highest-PR-AUC checkpoint, retaining the earlier checkpoint on ties. Its F1-optimal threshold is selected from the validation precision–recall curve in logit space to avoid sigmoid-saturation ties. Among tied F1 values, the first threshold in ascending order is chosen; scores at or above it are positive. The threshold is frozen before test evaluation, and test data do not determine retained samples or training budgets. Seed variability is reported as sample standard deviation and does not quantify uncertainty across splits or systems.

Reproducible records. Saved records include configurations, SHA-256 identities of input files, selected window indices, semantic and reference matrices, source code and checkpoints, training histories, and validation/test logits and labels in manifest order. Metric checks verify label order, threshold reuse, and confusion counts. The graph audit also retains means, medians, 95th percentiles, minima, and maxima, checks sequence-based graphs against the loader, and checks stored adjacencies for symmetry and binary entries. Hashes establish file identity, not annotation correctness. Deterministic PyTorch settings are requested with warnings for unsupported operations; bitwise equivalence across hardware is not claimed.

C Supplementary Transfer Results

C.1 Matched-Protocol Detection Results

Table 6 collects the aggregate detection results for the Full, Pre-align, and Target-only variants under Section B.4.

Table 6: Reported fixed-protocol HDFS↔OS results. Aggregate mean ± sample SD are reported for the Full, Pre-align, and Target-only variants.

Direction	Variant	F1 mean ± s.d.	PR-AUC mean ± s.d.	ROC-AUC mean ± s.d.
HDFS→OS	Full	0.9696 ± 0.0647	0.9004 ± 0.0090	0.9480 ± 0.0459
	Pre-align	0.9071 ± 0.0215	0.8807 ± 0.0211	0.9040 ± 0.0209
	Target-only	0.9537 ± 0.0042	0.8952 ± 0.0169	0.9226 ± 0.0494
OS→HDFS	Full	0.9960 ± 0.0070	0.999965 ± 0.000061	1.0000 ± 0.000002
	Pre-align	0.9955 ± 0.0078	0.998999 ± 0.001734	0.9990 ± 0.001715
	Target-only	0.9935 ± 0.0026	0.993797 ± 0.000005	0.994600 ± 0.001501

C.2 Matched-Budget Edge-Perturbation Analysis

High-mask-energy deletions produce larger mean absolute logit changes for both OpenStack classes and anomalous HDFS inputs; this ordering does not hold for normal HDFS inputs. The comparison

measures predictive sensitivity under matched edge-deletion budgets. Matching the number of deleted edges and the adjacency perturbation norm controls edit magnitude, but does not make the perturbations semantically equivalent.

Table 7: Mean absolute logit changes under matched-budget edge deletion, aggregated over three checkpoints and reported separately by sample label.

Direction	Label	High-mask energy	Low-mask energy	Random
HDFS→OS	Normal	.2793	.0995	.0861
HDFS→OS	Anomalous	.7249	.1234	.1598
OS→HDFS	Normal	.0066	.0076	.0060
OS→HDFS	Anomalous	.1105	.0277	.0223

High-mask-energy deletions produce larger mean logit changes for both OS classes and anomalous HDFS inputs; this ordering does not hold for normal HDFS inputs.

D Hyperparameter Sensitivity

Table 8 reports a separate BGL→TB sweep on a subset of 2,717 windows (2,508 normal and 209 anomalous), with validation-selected thresholds. Each sweep varies one hyperparameter within its fixed configuration. The reported target F1 values are distinct from the main-result protocol and are not averages across directions.

Table 8: Target F1 in a separate BGL→TB sensitivity diagnostic. L is the maximum temporal lag, λ_{anom} the few-label loss weight, γ the ranking margin, and λ_{align} the alignment weight.

(a) Maximum temporal lag order					
L	1	2	3	4	
F1	0.9298	0.9132	0.9071	0.9031	
(b) Few-label loss weight and margin					
λ_{anom}	0.02	0.05	0.10	0.20	
F1	0.9359	0.8557	0.9026	0.8972	
γ	0.02	0.05	0.08	0.20	
F1	0.9312	0.8267	0.8894	0.9115	
(c) Shared alignment weight					
λ_{align}	0	0.001	0.005	0.010	0.020
F1	0.8490	0.9045	0.9412	0.9132	0.9120

Temporal lag. F1 decreases across the tested lag orders, with $L = 1$ giving the best result. This supports a short within-band temporal context in this sweep, without identifying the mechanism behind the degradation at larger lags.

Few-label supervision. The smallest tested loss weight and ranking margin give the highest F1 in their respective sweeps. Increasing either parameter produces non-monotonic changes, so stronger few-label constraints do not yield consistent gains.

Alignment weight. Moderate alignment improves on the zero-weight setting, with the best tested F1 at $\lambda_{\text{align}} = 0.005$. Larger weights reduce F1. This demonstrates a setting-specific trade-off, rather than a universal benefit from stronger alignment or stability across seeds and domains.

E Efficiency and Computational Complexity

E.1 Band-Count Sweep

Table 9 evaluates inference as K varies on the fixed-size TB graph. Profiling uses batch size 1 and 500 timed normal-validation sequences. Peak allocated GPU memory is reported; latency excludes data loading, preprocessing, and initialization. The sweep does not measure end-to-end training cost, scaling across node counts, or efficiency relative to external baselines.

Table 9: Complexity, efficiency, and diagnostic detection performance with different numbers of graph-spectral bands under identical evaluation settings. Params: parameters (millions); Mem.: peak allocated memory (GB); Lat.: latency (ms/sequence); Thput.: throughput (sequences/s).

(a) Complexity and efficiency

Bands	Params (M)	Mem. (GB)	Lat. (ms)	Thput.
1	68.55	0.696	8.89	112.45
2	69.39	0.707	13.37	74.81
3	70.23	0.716	18.09	55.29

(b) Detection performance

Bands	ROC-AUC	F1	PR-AUC
1	0.988	**0.956**	0.927
2	0.993	0.917	0.859
3	**0.998**	0.923	**0.965**

Latency increases approximately linearly with the number of bands, while detection quality is not monotonic. The single-band variant has the highest F1 and lowest latency; three bands give the best ROC-AUC and PR-AUC; two bands trail on F1 and PR-AUC. Additional bands therefore trade latency and operating-point performance against ranking quality rather than producing a uniform improvement.

Profiling environment. The reported efficiency measurements use Ubuntu 24.04.4 LTS, an NVIDIA GeForce RTX 3090 with 24 GB memory, two Intel Xeon Silver 4214R CPUs, and 251 GiB system memory, with Python 3.12.11, PyTorch 2.10.0, and CUDA 12.8.

E.2 Dense Implementation Costs

Let d denote hidden width, with N, T, and K as defined above. Eigendecomposition of the full reference Laplacian via `torch.linalg.eigh` costs $O(N^3)$ time and $O(N^2)$ storage at initialization; the resulting basis is cached per domain. For each snapshot, feature projection $\mathbf{U}^\top\mathbf{H}$ costs $O(N^2d)$, dense adjacency projection $\mathbf{U}^\top\mathbf{A}\mathbf{U}$ costs $O(N^3)$, and Gaussian band weighting costs $O(KNd)$.

Dense spatial and temporal attention contribute $O(KTN^2d)$ and $O(KNT^2d)$ per sequence, giving the principal spectral/attention cost

$$O\big(TN^3 + TN^2d + KTNd + KTN^2d + KNT^2d\big)\,, \tag{46}$$

with additional GCN, feature-projection, and domain-specific TCN costs. For fixed head count, attention tensors require $O(KTN^2 + KNT^2)$ memory per sequence, beyond parameters, spectral features, and the cached basis; batch size multiplies activation costs. Training additionally stores backward activations and evaluates a dense matrix-exponential acyclicity penalty per band, conventionally cubic in N at fixed numerical accuracy.

Input sparsity does not eliminate these dense operations. Increasing K repeats band-specific attention while preserving the cached basis and scoring components shared within each domain. Sparse polynomial filtering or basis truncation could reduce cost, but neither is implemented or evaluated here.

F Spectral-Basis Sensitivity and Cross-Domain Comparison

This analysis covers BGL→TB, HDFS↔BGL, and HDFS↔OS. Each direction uses its own retained trained model; the HDFS↔OS diagnostics use the seed-42 full-model checkpoints. Band weights and shared representations are compared at fixed model parameters.

F.1 Band Preferences and Representation Discrepancies

Each domain contributes 64 nonoverlapping normal validation windows per sampling draw. The HDFS↔BGL and HDFS↔OS results use five draws, allowing windows to recur between

draws; BGL→TB retains its original single draw. Table 10 reports first-draw mean weights in low/middle/high band order. Table 11 reports band-weight Jensen–Shannon (JS) divergence in nats and biased maximum mean discrepancy (MMD^2) between the 64-dimensional shared representations. The RBF kernel's squared bandwidth is the median positive pooled squared distance from the first draw, held fixed for later draws within that direction.

Table 10: Source and target mean low-/middle-/high-frequency band weights in the first sampling draw.

Transfer direction	Source mean weights	Target mean weights
BGL→TB	$(0.02484, 0.97412, 0.00105)$	$(0.00020, 0.99977, 0.00004)$
HDFS→BGL	$(0.06866, 0.00248, 0.92886)$	$(0.00016, 0.99926, 0.00058)$
BGL→HDFS	$(0.97643, 0.01372, 0.00985)$	$(0.01076, 0.03677, 0.95247)$
HDFS→OS	$(0.63141, 0.05367, 0.31492)$	$(0.04100, 0.38190, 0.57710)$
OS→HDFS	$(0.77465, 0.20835, 0.01700)$	$(0.82436, 0.07018, 0.10546)$

Table 11: Fixed-checkpoint domain discrepancies: five-draw ranges, except for the original single-draw BGL→TB result (†). JS is in nats; MMD uses an RBF kernel.

Transfer direction	Band-weight JS	Biased MMD^2	Squared bandwidth
BGL→TB†	0.00848	0.791	7.785
HDFS→BGL	0.68140–0.68148	0.886–0.995	133.041
BGL→HDFS	0.62124–0.62158	0.796–0.797	86.365
HDFS→OS	0.24430–0.24657	0.907–0.919	71.697
OS→HDFS	0.03523–0.03651	0.795–0.796	88.231

BGL and TB both favor the middle band in BGL→TB. The HDFS↔BGL checkpoints exhibit different source/target preferences, as does HDFS→OS, where HDFS favors the low band and OS the high band. In OS→HDFS, both domains favor the low band, yet their shared representations retain a nonzero empirical discrepancy. Thus, similar band preferences do not establish representation alignment, and the preferences are properties of the evaluated directional checkpoints rather than fixed dataset characteristics. Absolute MMD values should not be ranked across directions because the learned representations and bandwidths differ.

F.2 Sensitivity to Spectral-Basis Changes

For each encoder domain, Table 12 evaluates 32 fixed validation windows under three sign flips and three rotations within numerically repeated eigenspaces. Entries give the range of mean absolute logit changes across these transformations.

Table 12: Basis sensitivity across the five transfer directions. Each range summarizes mean absolute logit changes over three transformations.

Transfer direction	Encoder domain	Sign flips	Eigenspace rotations
BGL→TB	BGL	3.003–3.534	0.0465–0.0915
BGL→TB	TB	0.137–5.109	3.2123–13.0034
HDFS→BGL	HDFS	0.014–0.330	0.0126–0.0300
HDFS→BGL	BGL	5.976–10.325	3.8369–5.9904
BGL→HDFS	BGL	4.052–4.708	0.0003–0.0010
BGL→HDFS	HDFS	1.352–33.572	0.1025–0.8757
HDFS→OS	HDFS	0.047–0.258	0.0030–0.0267
HDFS→OS	OS	1.699–4.873	0.2136–0.4091
OS→HDFS	OS	0.304–1.245	0.0229–0.0398
OS→HDFS	HDFS	0.626–18.358	0.0137–0.0218

Identity-control logit changes and reconstructed vertex-filter changes remain below 10^{-5}; restoring the original buffers recovers identical scores. Sign flips nevertheless alter outputs in every evaluated direction, demonstrating that the tested predictors are not basis-invariant. Near-repeated-eigenspace rotations provide additional numerical sensitivity evidence, with relative operator residuals below 10^{-5}. These logit changes do not measure F1 degradation or the performance of retrained models.

Interpretation across seeds and domains. The sampling ranges in Table 11 measure repeatability under data resampling, not training-seed stability. No matched independent-seed set was identified for the retained HDFS↔BGL models. Although the HDFS↔OS evaluation includes three independent seeds, the spectral diagnostics use only one; claims about stable band preferences or representations would require repeated-training discrepancies alongside F1/PR-AUC dispersion under fixed splits and supervision. Within-domain mask comparisons must resolve spectral-basis ambiguity, and direct element-wise comparisons across the four datasets are invalid because their vocabularies and spectral bases differ. D

G Figure and Table Abbreviations

Table notation. B/T/H denote BGL/TB/HDFS in Table 1, and arrows point from source to target. Avg. is the unweighted mean of the four displayed dataset or direction scores. Metric definitions are given in Section B.4; Table 2 uses PR-AUC, while Table 9 distinguishes PR-AUC and ROC-AUC. ΔF1 is $100(F1_{\text{variant}} - F1_{\text{full}})$ in percentage points; differences computed from rounded F1 values may not reproduce all reported decimals.

Ablation notation. w/o means without; Align., Reg., and Dyn. Fusion denote alignment, regularization, and dynamic band fusion. Multi-band and Freq. refer to spectral decomposition, and CC denotes constraint-aware screening. Base disables Freq. and CC, the single-component variants enable only the named component, and Full SCAST enables both. Pre-screen Align. denotes the alignment-placement control in Section B.4.

Model and vector notation. GCN, ST, TCN, and GRL denote graph convolutional network, spatial–temporal, temporal convolutional network, and gradient-reversal layer. "Dep." and "Can." abbreviate dependencies and candidate. In Figure 2, S/T denote source/target, B1–B3 denote low/middle/high spectral bands, Δ_k is the band-wise mean shift, and w_k is a fusion weight. O marks the origin and P1–P3 the endpoints of successive weighted vector additions. This is a schematic illustration, not an experimental result.

Diagnostic-figure notation. In Figure 3, blue, orange, and red bars denote screened, residual, and random perturbations, respectively. Assigned energy is the mean normalized candidate-edge perturbation energy in the designated band. Guided route is the norm of the screened-summary change divided by the sum of the screened- and residual-summary change norms in the extended diagnostic model; it is not a direct measure of dependency transferability. Anomaly-logit change is an absolute logit difference. The plotted "Target advantage" is the target-domain screened response minus the larger residual or random response. It is a sensitivity difference, not a detection-performance gain. Bars pool diagnostic records, whereas the advantage line uses target-domain records.