BEYOND A FIXED PATCH SCALE: SCALE-DECOUPLED MULTI-GRANULARITY EVIDENCE FUSION FOR TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Patch-based time-series anomaly detection typically relies on a predefined patch length, thereby restricting anomaly observation to a single temporal receptive field. However, real-world anomalies exhibit substantial heterogeneity in duration and contextual dependency, making a fixed scale prone to scale mismatch. To address this issue, we propose MGF-AD (Multi-Granularity Evidence Fusion for Scale-Uncertain Time-Series Anomaly Detection), which interprets patch length as the temporal granularity of anomaly observation. MGF-AD independently learns scale-specific normal patterns at multiple temporal scales, projects the resulting patch-level anomaly evidence onto a shared timeline via patch-to-point alignment, and performs per-scale score calibration followed by test-label-free fusion. This design explicitly decouples scale-specific normality modeling from cross-scale anomaly decision making. On TSB-AD-U and TSB-AD-M, MGF-AD achieves an overall advantage over the globally best fixed scale on the primary detection metrics and partially closes the performance gap between a globally fixed scale and the per-series optimal scale. Further analysis shows that anomaly evidence from different temporal granularities is complementary, suggesting that replacing fixed-scale selection with cross-scale evidence integration can reduce the sensitivity of patch-based TSAD to a single predefined temporal granularity.

1 INTRODUCTION

Time-series anomaly detection (TSAD) aims to identify anomalous timestamps or segments that deviate from normal dynamics in continuously observed time series. It has been widely applied to industrial monitoring, cloud service operations, and intelligent infrastructure. Since real-world anomalies are typically characterized by scarce annotations, severe class imbalance, and diverse patterns, unsupervised anomaly detection, as well as semi-supervised anomaly detection where only normal training data are available, has received increasing attention. In recent years, reconstruction-based models, probabilistic models, and attention-based approaches have continuously advanced the capability of temporal modeling Choi et al. (2021); Zamanzadeh Darban et al. (2024). Meanwhile, rigorous evaluation studies have shown that factors such as dataset quality, point adjustment, and threshold selection can substantially affect performance comparisons among different methods. It is therefore essential to validate model improvements under a unified and reliable evaluation protocol Liu & Paparrizos (2024).

Local temporal structures provide an important inductive bias for TSAD. PatchTST partitions a continuous time series into subseries-level patches and models them as Transformer tokens Nie et al. (2023). PaAno further extends patch-based representation learning to anomaly detection by learning patch representations with a lightweight local encoder and constructing a reference set from normal training patches for distance-based anomaly scoring Park & Kang (2026). However, existing patch-based methods typically require a predefined patch length, which imposes a fixed temporal receptive field for capturing local patterns. In practice, anomalies can vary substantially in both duration and contextual dependency. Shorter windows preserve fine-grained local variations but may lack sufficient temporal context, whereas longer windows capture more complete temporal structures but may dilute short-lived anomalous signals. Consequently, a single patch scale is inherently difficult to robustly accommodate anomalies with heterogeneous temporal structures.

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.

Motivated by this observation, we further interpret the patch length as the temporal granularity at which anomalies are observed, and define the mismatch between a fixed temporal granularity and the temporal context required to characterize an anomaly as scale mismatch. Since the optimal patch scale for each time series or anomalous segment is generally unknown at test time, a fixed single-scale model further suffers from scale uncertainty. To address this issue, we propose MGF-AD (Multi-Granularity Evidence Fusion for Scale-Uncertain Time-Series Anomaly Detection). MGF-AD first constructs overlapping local observations at multiple patch scales and independently learns normal local structures together with a scale-specific prototype bank of normal patterns for each scale. The patch-level anomaly scores produced at different scales are then mapped onto a unified timeline via coverage averaging, followed by scale-wise score calibration. Finally, point-level anomaly evidence from different temporal granularities is fused in a test-label-free and equally weighted manner. In this way, MGF-AD explicitly decouples scale-specific normality modeling from cross-scale anomaly decision making, thereby reducing the dependence of detection results on any single predefined patch scale.

Our main contributions are summarized as follows:

- We identify the scale-mismatch problem in patch-based TSAD. We reinterpret patch length as the temporal granularity of anomaly observation and reveal the limitations of fixed-scale selection from the perspectives of anomaly duration and the per-series optimal scale.
- We propose a multi-granularity, scale-decoupled normality modeling framework. Scale-specific normal local structures are learned independently at multiple patch scales, producing complementary anomaly evidence under different temporal contexts.
- We propose a calibrated point-level multi-scale anomaly evidence fusion mechanism. Through patch-to-point alignment, scale-wise score calibration, and test-label-free fusion, anomaly evidence from different temporal granularities is integrated on a unified timeline.

2 Related Work

Existing TSAD methods span reconstruction, probabilistic modeling, temporal association, and structural modeling Sakurada & Yairi (2014); Su et al. (2019); Xu et al. (2021); Yang et al. (2023); Kim et al. (2025); Dai et al. (2024), while recent studies also emphasize reliable evaluation protocols and range-aware metrics Liu & Paparrizos (2024); Paparrizos et al. (2022); Sarfraz et al. (2024). Patch-based models capture local temporal structure using subseries-level observations Nie et al. (2023); Park & Kang (2026); Grover et al. (2024), but their fixed patch length imposes a predefined temporal receptive field that may mismatch anomalies of different durations. Multi-scale TSAD methods alleviate this issue through multi-resolution representations Yang et al. (2023); Zhang et al. (2019); Wu et al. (2025); in contrast, MGF-AD independently models scale-specific normal patterns and performs patch-to-point alignment, per-scale calibration, and point-wise evidence fusion, shifting the focus from multi-scale representation learning to anomaly-evidence integration under scale uncertainty.[1]

3 Method

3.1 Problem Definition and Overall Framework

Given multivariate time series $\mathbf{X} \in \mathbb{R}^{B\times T\times C}$, where B, T, and C denote batch size, sequence length, and number of variables, let $\mathbf{x}_t \in \mathbb{R}^C$ be the observation at timestamp t. We consider the semi-supervised setting in which only normal intervals are used for training. For each test series, the goal is to produce point-wise anomaly scores

$$\mathbf{S} = [S_1, S_2, \ldots, S_T] \in \mathbb{R}^T, \tag{1}$$

where larger S_t indicates stronger deviation from learned normal patterns.

[1] A more detailed discussion of related TSAD, patch-based modeling, and multi-scale anomaly detection methods is provided in Appendix A.

MGF-AD (Multi-Granularity Evidence Fusion for Scale-Uncertain Time-Series Anomaly Detection) comprises multi-scale patch construction, scale-decoupled normality modeling, and calibrated point-wise evidence fusion:

$$\mathbf{X} \to \{\mathbf{P}^{(m)}\}_{m=1}^{M} \to \{\mathbf{a}^{(m)}\}_{m=1}^{M} \to \{\mathbf{s}^{(m)}\}_{m=1}^{M} \to \{\tilde{\mathbf{s}}^{(m)}\}_{m=1}^{M} \to \mathbf{S}, \tag{2}$$

where M is the number of scales, and $\mathbf{P}^{(m)}$, $\mathbf{a}^{(m)}$, $\mathbf{s}^{(m)}$, and $\tilde{\mathbf{s}}^{(m)}$ denote the patch sequence, patch-level scores, aligned point-wise scores, and calibrated evidence at scale m, respectively. Figure 1 illustrates the full framework.

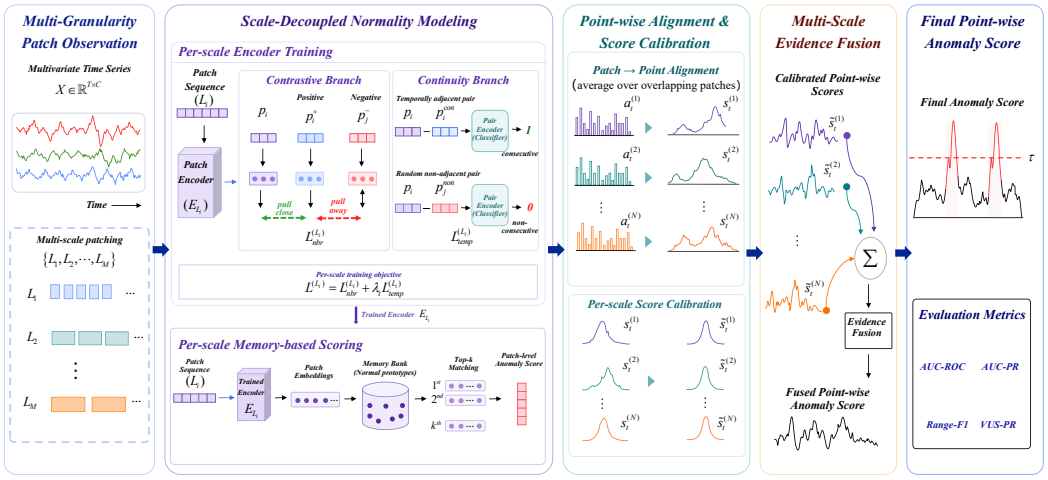


Figure 1: Overview of MGF-AD. Scale-specific branches learn normal patterns and produce patch-level anomaly scores, which are aligned to the original timeline, calibrated, and fused into point-wise anomaly scores.

3.2 Multi-Granularity Patch Observation

Let $\mathcal{L} = \{L_1, \ldots, L_M\}$ denote the candidate patch lengths, where smaller L_m captures local variations and larger L_m provides broader temporal context. At scale m, patches are extracted with stride ρ and start index $\tau_i = 1 + (i-1)\rho$:

$$\mathbf{p}_i^{(m)} = [\mathbf{x}_{\tau_i}, \ldots, \mathbf{x}_{\tau_i + L_m - 1}] \in \mathbb{R}^{L_m \times C}. \tag{3}$$

This yields $N_m = \lfloor (T - L_m)/\rho \rfloor + 1$ patches. We use $\rho = 1$ throughout, and denote the resulting scale-specific patch sequence by $\mathbf{P}^{(m)} = [\mathbf{p}_1^{(m)}, \ldots, \mathbf{p}_{N_m}^{(m)}]$.

3.3 Scale-Decoupled Normality Modeling

Each scale learns an independent normality model and outputs a patch anomaly sequence $\mathbf{a}^{(m)}$.

Scale-specific patch representation. At scale m, encoder f_{θ_m} and projection head g_{ϕ_m} produce

$$\mathbf{h}_i^{(m)} = f_{\theta_m}(\mathbf{p}_i^{(m)}) \in \mathbb{R}^d, \tag{4}$$

$$\mathbf{z}_i^{(m)} = g_{\phi_m}(\mathbf{h}_i^{(m)}) \in \mathbb{R}^{d_z}, \tag{5}$$

where d and d_z are the embedding and projection dimensions. All scales share the same architecture type but use independent parameters $\{\theta_m, \phi_m\}$.

Neighborhood-consistency metric learning. For anchor i, define

$$\Omega_r^{(m)}(i) = \{j \mid 0 < |j - i| \le r,\ 1 \le j \le N_m\}, \tag{6}$$

where r is the positive-sampling radius. We sample $i^+ \in \Omega_r^{(m)}(i)$ and a non-matching i^- from the mini-batch. With cosine distance

$$d_{\cos}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}, \tag{7}$$

the neighborhood-consistency loss is

$$\mathcal{L}_{\text{nbr}}^{(m)} = \frac{1}{B_p} \sum_{i=1}^{B_p} \max\Big(0, d_{\cos}(\mathbf{z}_i^{(m)}, \mathbf{z}_{i^+}^{(m)}) - d_{\cos}(\mathbf{z}_i^{(m)}, \mathbf{z}_{i^-}^{(m)}) + \delta\Big), \tag{8}$$

where B_p is the patch mini-batch size and δ the margin.

Temporal-continuity regularization. Adjacent and randomly sampled non-adjacent patch pairs are labeled by $y_{ij}^{(m)} \in \{0, 1\}$. A scale-specific classifier predicts

$$\pi_{ij}^{(m)} = \sigma\Big(c_{\psi_m}([\mathbf{h}_i^{(m)}; \mathbf{h}_j^{(m)}])\Big), \tag{9}$$

where $[\cdot;\cdot]$ denotes concatenation and σ the sigmoid function. Over pair set $\mathcal{Q}_m$, the auxiliary loss is

$$\mathcal{L}_{\text{temp}}^{(m)} = -\frac{1}{|\mathcal{Q}_m|} \sum_{(i,j)\in\mathcal{Q}_m} \Big[y_{ij}^{(m)} \log \pi_{ij}^{(m)} + (1 - y_{ij}^{(m)}) \log(1 - \pi_{ij}^{(m)})\Big]. \tag{10}$$

The scale-specific objective is

$$\mathcal{L}^{(m)} = \mathcal{L}_{\text{nbr}}^{(m)} + \lambda_t \mathcal{L}_{\text{temp}}^{(m)}, \tag{11}$$

with

$$\lambda_t = \lambda_0 \max\left(0, 1 - \frac{t}{\eta I}\right), \tag{12}$$

where t is the current iteration, I the total number of iterations, λ_0 the initial auxiliary-loss weight, and $\eta \in (0, 1]$ the active training fraction. Thus $\lambda_t = 0$ for $t \geq \eta I$.

Scale-specific normal prototype bank. After training, only f_{θ_m} is retained for anomaly scoring. Normal training embeddings are summarized into a scale-specific prototype bank

$$\mathcal{M}^{(m)} = \{\boldsymbol{\mu}_1^{(m)}, \ldots, \boldsymbol{\mu}_{R_m}^{(m)}\}, \tag{13}$$

where $\boldsymbol{\mu}_r^{(m)} \in \mathbb{R}^d$ denotes a normal prototype and R_m is the bank size. We construct the bank using MiniBatch k-means on L_2-normalized normal embeddings and retain, for each cluster, the training embedding nearest to its centroid as the representative prototype. The bank size is set to 10% of the normal training patches.

For a test embedding $\mathbf{h}_i^{(m)}$, the patch anomaly score is

$$a_i^{(m)} = 1 - \max_{\boldsymbol{\mu}\in\mathcal{M}^{(m)}} \cos(\mathbf{h}_i^{(m)}, \boldsymbol{\mu}), \tag{14}$$

corresponding to nearest-prototype scoring with $k = 1$.

3.4 Calibrated Point-wise Multi-Scale Evidence Fusion

Because $\mathbf{a}^{(m)}$ is patch-level and scale dependent, cross-scale evidence must first be aligned to timestamps and calibrated before fusion.

Patch-to-point alignment. For timestamp t, let

$$\mathcal{C}_t^{(m)} = \{i \mid 1 \leq i \leq N_m,\ \tau_i \leq t \leq \tau_i + L_m - 1\} \tag{15}$$

be the indices of scale-m patches covering t. Their scores are averaged as

$$s_t^{(m)} = \frac{1}{|\mathcal{C}_t^{(m)}|} \sum_{i\in\mathcal{C}_t^{(m)}} a_i^{(m)}, \tag{16}$$

yielding the aligned sequence $\mathbf{s}^{(m)} = [s_1^{(m)}, \ldots, s_T^{(m)}]^\top$.

Per-scale score calibration. Each aligned sequence is independently calibrated,

$$\tilde{s}_t^{(m)} = \Gamma^{(m)}(s_t^{(m)}), \tag{17}$$

where $\Gamma^{(m)}$ denotes the calibration operator. The main experiments use per-sequence min–max normalization,

$$\tilde{s}_t^{(m)} = \frac{s_t^{(m)} - s_{\min}^{(m)}}{s_{\max}^{(m)} - s_{\min}^{(m)} + \epsilon}, \tag{18}$$

with $s_{\min}^{(m)} = \min_t s_t^{(m)}$, $s_{\max}^{(m)} = \max_t s_t^{(m)}$, and $\epsilon > 0$ for numerical stability. Because the extrema are computed over the evaluated sequence, calibration is performed offline per sequence.

Multi-scale evidence fusion. The final score is the equal-weight average

$$S_t = \frac{1}{M} \sum_{m=1}^{M} \tilde{s}_t^{(m)}, \tag{19}$$

which introduces neither learned scale weights nor test-label supervision. Applying Eq. 19 to all timestamps yields the final score sequence $\mathbf{S}$ defined in Eq. 1. Ground-truth labels are used only for evaluation.

4 Experiments

TSB-AD benchmark. We conduct experiments on the TSB-AD benchmark, using the Tuning and Eval splits of both the univariate TSB-AD-U and multivariate TSB-AD-M collections. The Tuning split is used exclusively for hyperparameter selection, while the Eval split is reserved for final performance reporting. Dataset statistics are summarized in Table 1.

Table 1: Statistics of the TSB-AD-U and TSB-AD-M datasets used in our experiments.

Category	Split	# Time-Series	Avg. Length	Anomaly Ratio
TSB-AD-U	Tuning	48	47143.3	3.5%
	Eval	350	51886.7	4.5%
TSB-AD-M	Tuning	20	98164.1	5.7%
	Eval	180	108826.7	5.0%

Implementation details. MGF-AD uses three patch scales, $\mathcal{L} = \{64, 96, 128\}$, with an independently parameterized four-layer 1D-CNN encoder and non-affine RevIN at each scale. The resulting anomaly evidence is fused with equal weights after alignment and per-scale calibration. Full architecture and training settings are provided in Appendix C.1 and Table 11. Unless otherwise specified, single-run results use seed 2027, while repeated experiments are explicitly reported as mean $\pm$ standard deviation.

Baselines. We compare MGF-AD with representative statistical, neural-network-based, Transformer-based, and time-series foundation-model baselines, as summarized in Tables 2– 3. We additionally include PaAno Park & Kang (2026), a closely related patch-based anomaly detection method. To isolate the effect of multi-scale modeling, we construct three fixed-scale variants, Scale-64, Scale-96, and Scale-128, which share the same architecture, training objectives, prototype bank construction, and anomaly scoring procedure as MGF-AD, but use only a single patch scale. They serve as controlled baselines for evaluating the gain from multi-scale anomaly evidence fusion.

Metrics. We report VUS-PR, VUS-ROC, Range-F1, AUC-PR, AUC-ROC, and Point-F1 without point adjustment, with VUS-PR as the primary metric. All metrics are computed independently for each test series and macro-averaged over the Eval split. Detailed metric definitions and the macro-averaging protocol are provided in Appendix D.

4.1 Results and Discussion

Tables 2 and 3 report the performance of MGF-AD and representative statistical, neural, Transformer-based, and time-series foundation-model baselines on TSB-AD-U and TSB-AD-M, respectively. Overall, MGF-AD performs consistently well on both benchmarks, particularly on VUS-PR and Range-F1.

On TSB-AD-U, MGF-AD achieves the best results on five of six metrics and matches the strongest baseline on AUC-ROC. Compared with PaAno, the strongest external baseline overall, MGF-AD further improves VUS-PR, Range-F1, AUC-PR, and Point-F1, with the largest gain on Range-F1.

Several simpler statistical and machine-learning methods remain competitive on individual metrics, while larger Transformer-based and foundation models show no consistent advantage.

Table 2: Overall anomaly detection performance on TSB-AD-U. Results are reported without point adjustment; higher values indicate better performance.

	Method	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1	# Params	Run Time
Stat&ML	(Sub)-PCA (2016)	0.42	0.76	0.41	0.37	0.71	0.42	-	1.5s
	KShapeAD (2017)	0.40	0.76	0.40	0.35	0.74	0.39	-	8.0s
	DLinear (2023)	0.27	0.76	0.24	0.23	0.67	0.28	$< 0.1M$	2.9s
	NLinear (2023)	0.26	0.74	0.21	0.20	0.63	0.24	$< 0.1M$	5.8s
NN	DeepAnT (2018)	0.34	0.79	0.35	0.33	0.71	0.38	$< 0.1M$	2.0s
	USAD (2020)	0.36	0.71	0.40	0.32	0.66	0.37	$< 0.1M$	1.7s
	TimesNet (2023)	0.26	0.72	0.21	0.18	0.61	0.24	$< 0.1M$	11.2s
	FITS (2024)	0.26	0.73	0.20	0.17	0.64	0.23	$< 0.1M$	3.1s
	DADA (2025)	0.31	0.77	0.31	0.29	0.71	0.33	1.84M	0.8s
	KAN-AD (2025)	0.40	0.82	0.43	0.41	0.80	0.44	$< 0.1M$	12.1s
	PaAno (2026)	0.53	0.89	0.49	0.47	**0.87**	0.52	0.3M	5.4s
Transformer	AnomalyTransformer (2021)	0.12	0.56	0.14	0.08	0.50	0.12	4.8M	48.9s
	DCdetector (2023)	0.09	0.57	0.07	0.05	0.50	0.10	0.8M	5.8s
	Lag-Llama (2023)	0.27	0.72	0.31	0.25	0.65	0.30	2.5M	1220.8s
	OFA (2023)	0.24	0.71	0.20	0.16	0.59	0.22	81.9M	171.1s
	PatchTST (2023)	0.30	0.76	0.25	0.23	0.65	0.27	6.9M	26.3s
	iTransformer (2024)	0.32	0.77	0.22	0.21	0.67	0.26	6.3M	9.8s
	MOMENT (FT) (2024)	0.39	0.76	0.35	0.30	0.69	0.35	109.6M	43.6s
	MOMENT (ZS) (2024)	0.38	0.75	0.36	0.30	0.68	0.35	109.6M	42.9s
	TimesFM (2024)	0.30	0.74	0.34	0.28	0.67	0.34	203.5M	83.8s
	MGF-AD (ours)	**0.54**	**0.90**	**0.51**	**0.48**	**0.87**	**0.53**	0.87M	14.6s

On TSB-AD-M, MGF-AD achieves the best result on all six metrics. Relative to PaAno, the largest gains again occur on VUS-PR and Range-F1, with additional improvements on VUS-ROC, AUC-PR, AUC-ROC, and Point-F1. The gains across multiple metric families indicate that the improvement is not tied to a single evaluation criterion.

Table 3: Overall anomaly detection performance on TSB-AD-M. Results are reported without point adjustment; higher values indicate better performance.

	Method	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1	# Params	Run Time
Stat&ML	KMeansAD (2001)	0.29	0.73	0.33	0.25	0.69	0.31	-	62.0s
	PCA (2016)	0.31	0.74	0.29	0.31	0.70	0.37	-	0.1s
	DLinear (2023)	0.29	0.70	0.21	0.27	0.66	0.32	$< 0.1M$	14.8s
	NLinear (2023)	0.30	0.70	0.21	0.26	0.65	0.31	$< 0.1M$	15.0s
NN	DeepAnT (2018)	0.31	0.76	0.37	0.32	0.73	0.37	$< 0.1M$	9.5s
	OmniAnomaly (2019)	0.31	0.69	0.37	0.27	0.65	0.32	$< 0.1M$	9.1s
	TimesNet (2023)	0.19	0.64	0.17	0.13	0.56	0.20	$< 0.1M$	52.1s
	FITS (2024)	0.21	0.66	0.16	0.15	0.58	0.22	$< 0.1M$	16.7s
	DADA (2025)	0.31	0.73	0.25	0.31	0.69	0.35	1.84M	2.1s
	KAN-AD (2025)	0.40	0.76	0.41	0.35	0.73	0.40	$< 0.1M$	31.9s
	PaAno (2026)	0.43	0.79	0.41	0.38	0.76	0.43	0.3M	9.7s
Transformer	AnomalyTransformer (2021)	0.12	0.57	0.14	0.07	0.52	0.12	4.8M	55.8s
	DCdetector (2023)	0.10	0.54	0.09	0.05	0.48	0.10	0.8M	15.0s
	OFA (2023)	0.21	0.63	0.17	0.15	0.55	0.21	81.9M	532.9s
	PatchTST (2023)	0.30	0.71	0.22	0.28	0.67	0.33	6.9M	66.9s
	iTransformer (2024)	0.30	0.71	0.18	0.24	0.64	0.29	6.4M	24.4s
	CATCH (2025)	0.30	0.72	0.18	0.23	0.65	0.29	210.8M	40.1s
	MGF-AD (ours)	**0.45**	**0.81**	**0.44**	**0.39**	**0.77**	**0.44**	0.94M	32.1s

These results motivate the ablations below, which examine the contributions of scale configuration and cross-scale evidence processing. For source-level variation, complete subset results for TSB-AD-U and TSB-AD-M are reported in Appendix Tables 7 and 8, respectively; the observed variation across subsets complements the overall macro-averaged results with a more fine-grained view of detection difficulty.

4.2 Ablation Study

We ablate four design factors: patch-scale configuration, multi-scale versus homogeneous ensembling, fusion strategy, and score normalization. Unless otherwise stated, MGF-AD uses scales $\{64, 96, 128\}$ with per-scale min–max normalization and arithmetic-mean fusion.

Patch-scale configuration. Fixed scales exhibit different preferences across datasets and metrics. VUS-PR improves from 0.4866/0.4389 for Scale-96 to 0.5417/0.4451 for $\{64, 96, 128\}$ on TSB-AD-U/TSB-AD-M, respectively. Since adding scales does not yield monotonic gains, performance depends on scale composition rather than scale count alone (Table 4).

Table 4: Ablation of patch-scale configurations on TSB-AD-U and TSB-AD-M. The default MGF-AD configuration $\{64, 96, 128\}$ is marked separately, and the best result for each metric is highlighted in bold.

Dataset	Patch Scales	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U	$\{32\}$	0.4467	0.8737	0.4479	0.3550	0.8387	0.4152
	$\{64\}$	0.4804	0.8825	0.4622	0.3914	0.8594	0.4465
	$\{96\}$	0.4866	0.8804	0.4582	0.4080	0.8619	0.4603
	$\{128\}$	0.4576	0.8745	0.4362	0.3809	0.8580	0.4376
	$\{32, 64\}$	0.4857	0.8867	0.4859	0.3981	0.8606	0.4514
	$\{64, 96\}$	0.5002	0.8870	0.4775	0.4182	0.8675	0.4690
	$\{96, 128\}$	0.4811	0.8803	0.4549	0.4052	0.8642	0.4591
	$\{32, 64, 96\}$	0.5097	0.8903	0.5032	0.4232	0.8686	0.4735
	$\{\mathbf{64}, \mathbf{96}, \mathbf{128}\}$	**0.5417**	**0.8958**	0.5077	**0.4846**	**0.8745**	**0.5310**
	$\{32, 64, 96, 128\}$	0.5155	0.8925	**0.5104**	0.4299	0.8734	0.4816
TSB-AD-M	$\{32\}$	0.3671	0.7619	0.3363	0.3158	0.7092	0.3615
	$\{64\}$	0.4172	0.7929	0.4023	0.3633	0.7510	0.4098
	$\{96\}$	0.4389	0.8051	0.4261	0.3876	0.7709	0.4386
	$\{128\}$	0.4141	0.7910	0.3980	0.3662	0.7619	0.4208
	$\{32, 64\}$	0.4107	0.7857	0.4054	0.3582	0.7401	0.4027
	$\{64, 96\}$	0.4397	0.8059	0.4339	0.3865	0.7690	0.4369
	$\{96, 128\}$	0.4399	0.8002	0.4325	0.3891	0.7690	0.4423
	$\{32, 64, 96\}$	0.4332	0.8007	0.4329	0.3821	0.7603	0.4267
	$\{\mathbf{64}, \mathbf{96}, \mathbf{128}\}$	0.4451	**0.8065**	0.4412	0.3928	**0.7725**	**0.4434**
	$\{32, 64, 96, 128\}$	**0.4471**	0.8060	**0.4445**	**0.3946**	0.7688	0.4414

Multi-scale versus homogeneous ensemble. To separate scale diversity from generic ensembling, we compare MGF-AD $\{64, 96, 128\}$ with a homogeneous 3×Scale-96 ensemble under identical procedures and matched branch counts. Both are evaluated over seeds 2025, 2026, and 2027, with results reported as mean ± standard deviation.

Table 5 shows that the heterogeneous configuration outperforms the homogeneous 3×Scale-96 ensemble across all six metrics on both datasets. Under matched branch counts, this suggests that scale diversity provides complementary anomaly evidence beyond generic ensembling, although ensemble-related effects cannot be fully excluded.

Cross-scale evidence processing. As shown in Table 6, we compare fusion strategies with min–max normalization fixed, and normalization strategies with arithmetic-mean fusion fixed. The results exhibit some dataset dependence. Considering the Tuning-split results and implementation simplicity, we adopt arithmetic-mean fusion with min–max normalization as a unified parameter-free default, fixed before evaluation on the Eval split.

4.3 Mechanistic Analysis of Multi-Granularity Fusion

We next analyze how fixed-scale preference varies with anomaly duration; representative scale-specific responses and cross-scale disagreement are provided in Appendix Figures 4–5.

Table 5: Multi-scale versus homogeneous ensemble control on TSB-AD-U and TSB-AD-M. Results are reported as mean ± standard deviation over three random seeds. $3 \times$ *Scale-96* uses three independently trained branches with the same patch scale, whereas MGF-AD uses heterogeneous scales $\{64, 96, 128\}$. Best results within each dataset are highlighted in bold.

	Dataset Configuration	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U	$3 \times Scale - 96$	0.4823±0.0035	0.8811±0.0014	0.4584±0.0034	0.402±0.0063	0.8627±0.0017	0.4567±0.0053
	$\{64, 96, 128\}$	0.5435±0.0016	0.8952±0.0014	0.5084±0.0006	0.4836±0.0016	0.8774±0.0021	0.5332±0.0017
TSB-AD-M	$3 \times Scale - 96$	0.4371±0.0009	0.8030±0.0046	0.4246±0.0038	0.3844±0.0022	0.7677±0.0056	0.4347±0.0026
	$\{64, 96, 128\}$	0.4425±0.0034	0.8060±0.0008	0.4399±0.0010	0.3895±0.0033	0.7715±0.0009	0.4396±0.0031

Table 6: Cross-scale evidence-processing ablation on TSB-AD-U and TSB-AD-M: fusion strategies (left) and score normalization (right). Best results within each dataset and ablation block are highlighted in bold.

Dataset	Fusion	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1	Normalization	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U	Mean	**0.5417**	**0.8958**	**0.5077**	**0.4846**	**0.8745**	**0.5310**	Min–max	**0.5417**	**0.8958**	**0.5077**	**0.4846**	**0.8745**	**0.5310**
	Max	0.4776	0.8848	0.4483	0.3969	0.8660	0.4547	Z-score	0.5040	0.8877	0.4806	0.4210	0.8704	0.4733
	Weighted	0.5006	0.8860	0.4752	0.4196	0.8688	0.4722	None	0.5028	0.8872	0.4814	0.4196	0.8696	0.4715
TSB-AD-M	Mean	0.4451	**0.8065**	0.4412	0.3928	**0.7725**	0.4434	Min–max	0.4451	0.8065	0.4412	0.3928	0.7725	0.4434
	Max	0.4187	0.7977	0.4105	0.3654	0.7631	0.4214	Z-score	**0.4484**	**0.8089**	**0.4479**	**0.3960**	**0.7747**	**0.4466**
	Weighted	**0.4453**	0.8054	**0.4441**	**0.3935**	0.7714	**0.4443**	None	0.4482	0.8069	0.4432	0.3949	0.7728	0.4456

Anomaly-duration sensitivity. Because patch length determines the temporal receptive field, we examine the sensitivity of different scales to anomaly duration. For the i-th test sequence, let

$$\mathcal{A}_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,K_i}\} \tag{20}$$

denote the contiguous anomalous segments extracted from the ground-truth labels, where K_i is the number of segments. We define the representative anomaly duration as the median segment length:

$$d_i = \text{median}\left(|A_{i,1}|, |A_{i,2}|, \ldots, |A_{i,K_i}|\right), \tag{21}$$

where $|A_{i,j}|$ is the length of the j-th anomalous segment. Test sequences are partitioned into Short, Medium, and Long groups according to the terciles of d_i. Ground-truth labels are used only for this post-hoc analysis and are not used by MGF-AD during inference or cross-scale fusion.

Figure 2 shows that fixed-scale preference varies with anomaly duration. On TSB-AD-U, different duration groups favor different fixed scales, whereas Fusion remains competitive across all groups. TSB-AD-M exhibits a similar but non-monotonic dependence, with Fusion achieving or approaching the best fixed-scale performance across duration groups. These results indicate that no fixed scale is uniformly optimal and that multi-scale fusion reduces sensitivity to a predefined temporal granularity.

4.4 Oracle-Gap Recovery and Computational Efficiency

Because a globally optimal fixed scale can remain suboptimal for individual sequences, we quantify how much of this scale-selection loss is recovered by multi-scale fusion and report its computational cost.

Best-single oracle-gap analysis. We define Best Fixed as the fixed scale with the highest macro-averaged performance on the Eval split, whereas Best-single Oracle retrospectively selects the best fixed scale for each test sequence. Gap recovery is quantified as $R_{\text{gap}} = (F - B)/(O - B) \times 100\%$, where B, F, and O denote Best Fixed, Fusion, and Oracle performance, respectively. Complete results are provided in Appendix B.3 and Table 9.

As shown in Figure 3, MGF-AD consistently recovers part of the gap without test-label-based scale selection, with recovery ranging from 11.81% to 40.46%.

Runtime and computational cost. On a single NVIDIA A800 80GB GPU, the serial Fusion implementation costs 2.97× and 3.16× the fastest fixed-scale variant on TSB-AD-U and TSB-AD-M, respectively. Scale-specific branches are independent and can be parallelized; full results are provided in Appendix Table 10.

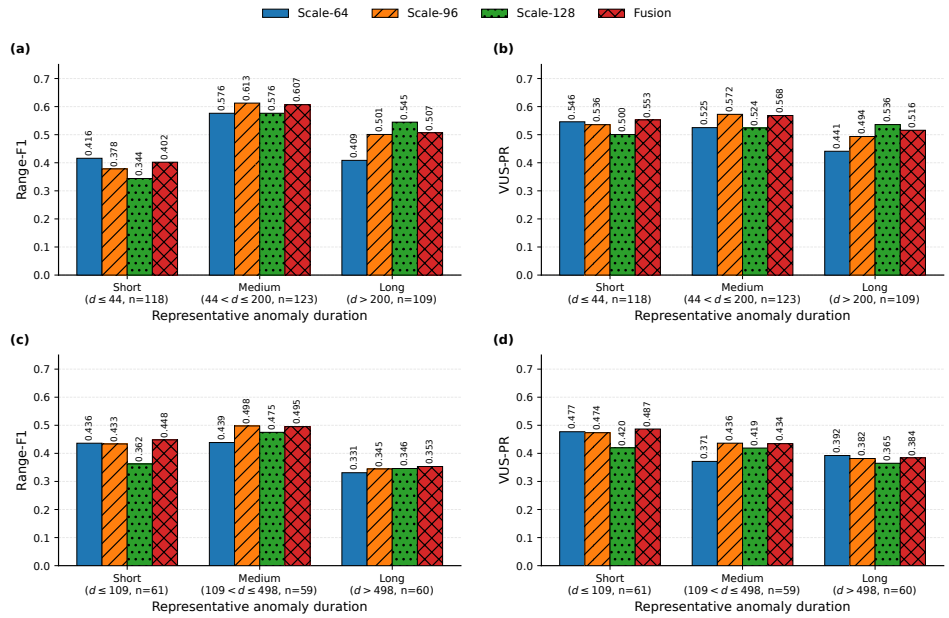


Figure 2: Performance across anomaly-duration groups on TSB-AD-U and TSB-AD-M. (a,b) Range-F1 and VUS-PR on TSB-AD-U; (c,d) the corresponding results on TSB-AD-M. Duration groups are defined by terciles of the median length of contiguous anomalous segments. Group boundaries and sample sizes are shown on the x-axis.

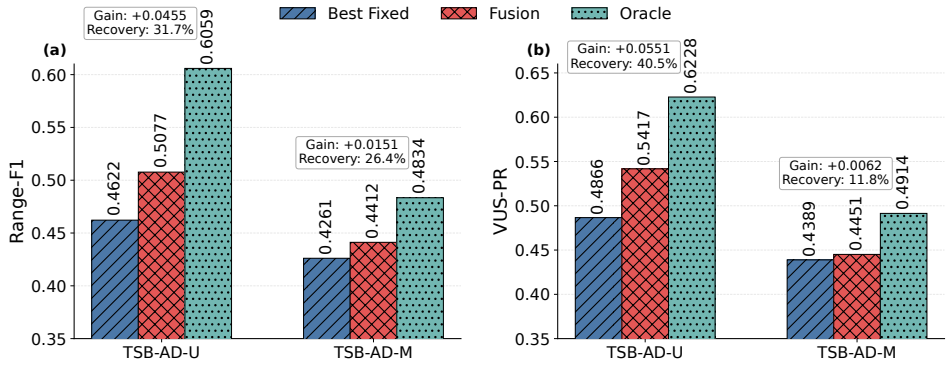


Figure 3: Best-single oracle-gap analysis on TSB-AD-U and TSB-AD-M. (a) Range-F1 and (b) VUS-PR. Best Fixed is the globally best fixed scale on the Eval split, whereas Oracle retrospectively selects the best fixed scale for each test sequence.

5 Conclusion

This work proposes MGF-AD to address scale uncertainty in patch-based time-series anomaly detection. By learning scale-specific normal patterns at multiple patch lengths and aligning, calibrating, and fusing their anomaly responses, MGF-AD reduces reliance on any single predefined scale. Experiments on TSB-AD-U and TSB-AD-M show that fixed-scale performance varies with anomaly duration, whereas multi-scale fusion reduces scale-selection loss without test-label-based scale selection. MGF-AD still requires predefined candidate scales, relies on sequence-level statistics for offline calibration, and incurs additional computation from independent scale branches. Future work will explore adaptive scale selection or weighting, online calibration using training-time normal statistics, and parameter sharing or parallel execution for more efficient deployment.

AI USE STATEMENT

Generative AI tools were used solely to assist with manuscript translation and language polishing. All research ideas, methodological design, experiments, data analysis, figures, tables, and reported results were produced and verified by the authors. The authors reviewed all AI-assisted text and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the complete formulation of MGF-AD in Section 3.1 and the experimental setup and evaluation protocol in Section 4. Detailed model architectures, training hyperparameters, prototype-bank settings, random-seed protocols, evaluation metrics, and macro-averaging procedures are provided in Appendix C and Appendix D. Additional subset-level results, mechanism analyses, oracle-gap details, and computational-cost measurements are reported in the appendix to facilitate independent reproduction of the reported results.

REFERENCES

Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pp. 1–34. Springer, 2016.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Kukjin Choi, Jihun Yi, Changhwa Park, and Sungroh Yoon. Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE access*, 9:120043–120065, 2021.

Zhihao Dai, Ligang He, Shuang-Hua Yang, and Matthew Leeke. Sarad: Spatial association-aware anomaly detection and diagnosis for multivariate time series. *Advances in Neural Information Processing Systems*, 37:48371–48410, 2024.

Abhimanyu Das, Wei-Cheng Kong, Rahul Sen, and Yizhou Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the International Conference on Machine Learning*, pp. 10148–10167, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *Proceedings of the International Conference on Machine Learning*, pp. 16115–16152, 2024.

Shivam Grover, Amin Jalali, and Ali Etemad. Segment, shuffle, and stitch: A simple layer for improving time-series representations. *Advances in Neural Information Processing Systems*, 37: 4878–4905, 2024.

Hyungi Kim, Jisoo Mok, Dongjun Lee, Jaihyun Lew, Sungjae Kim, and Sungroh Yoon. Causality-aware contrastive learning for robust multivariate time-series anomaly detection. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30591–30608, 2025.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE access*, 7:1991–2005, 2018.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

John Paparrizos and Luis Gravano. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)*, 42(2):1–49, 2017.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *Proc. VLDB Endow.*, 15(11):2774–2787, 2022.

Jinju Park and Seokho Kang. Paano: Patch-based representation learning for time-series anomaly detection. In *International Conference on Learning Representations*, 2026.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pp. 4–11, 2014.

M. Saquib Sarfraz, Mei-Yen Chen, Lukas Layer, Kunyu Peng, and Marios Koulakis. Position: Quo vadis, unsupervised time series anomaly detection? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43461–43476, 2024.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *International Conference on Learning Representations*, volume 2025, pp. 81358–81381, 2025.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *Proceedings of the International Conference on Learning Representations*, 2021.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. Fault detection by mining association rules from house-keeping data. In *Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, volume 3, 2001.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3033–3045, 2023.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):1–42, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1409–1416, 2019.

Quan Zhou, Changhua Pei, Fei Sun, Han Jing, Zhengwei Gao, Haiming Zhang, Gaogang Xie, Dan Pei, and Jianhui Li. Kan-ad: Time series anomaly detection with kolmogorov-arnold networks. In *International Conference on Machine Learning*, 2025.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Related Work

A.1 Time-Series Anomaly Detection and Evaluation

Time-series anomaly detection has gradually evolved from traditional statistical learning methods to deep neural networks and Transformer-based approaches. Representative studies include reconstruction-error-based anomaly detection methods Sakurada & Yairi (2014), OmniAnomaly, which combines stochastic latent variables with recurrent structures Su et al. (2019), Anomaly Transformer, which models temporal associations for anomaly detection Xu et al. (2021), and DCdetector, which leverages multi-scale dual attention and contrastive learning Yang et al. (2023). For multivariate time series, recent studies have further emphasized structural dependencies across variables. CAROTS incorporates causal relationships into contrastive learning to improve detection robustness Kim et al. (2025), while SARAD performs anomaly detection and diagnosis by modeling dynamic spatial associations among variables Dai et al. (2024). Collectively, these methods enhance anomaly representation from complementary perspectives, including reconstruction, probabilistic modeling, temporal association, and inter-variable dependency modeling.

Beyond model design, reliable evaluation protocols constitute another important aspect of TSAD research. Recent studies have highlighted the importance of simple baselines, non-trivial datasets, and rigorous evaluation protocols for obtaining reliable performance comparisons Sarfraz et al. (2024). TSB-AD further provides a systematic analysis of existing datasets, assumptions regarding anomaly distributions, and potential evaluation biases, while establishing a unified benchmark for TSAD Liu & Paparrizos (2024). To address the limitations of point-wise metrics in evaluating anomalous segments, VUS integrates detection thresholds with temporal tolerance ranges and introduces range-aware metrics such as VUS-ROC and VUS-PR Paparrizos et al. (2022). In this work, we follow the evaluation setting of TSB-AD and avoid the use of point adjustment.

A.2 Patch-Based Local Temporal Modeling

Patch-based modeling partitions a continuous time series into local subsequences, enabling the model to explicitly capture local dynamics within a limited temporal range. PatchTST treats subseries-level patches as Transformer tokens for temporal modeling Nie et al. (2023). S3 further models temporal dependencies at different granularities through temporal segmentation and learnable rearrangement Grover et al. (2024). In the context of anomaly detection, PaAno introduces patch-based representation learning into TSAD and performs anomaly scoring by combining local representation learning with a reference set of normal patches Park & Kang (2026). These studies demonstrate that local temporal structures and their observation granularities play an important role in time-series representation learning.

However, the patch length also determines the local temporal receptive field of the model, making a fixed patch scale difficult to accommodate anomalies with heterogeneous durations and contextual requirements. Unlike prior work that primarily focuses on local representation learning under a predefined scale, this work further investigates multi-granularity normality modeling and anomaly evidence integration when the appropriate scale cannot be determined in advance. Accordingly, MGF-AD retains patches as the basic units for local observation, while shifting the focus from fixed-scale representation learning to multi-scale anomaly decision making under scale uncertainty.

A.3 Multi-Scale Modeling and Anomaly Evidence Fusion

Multi-scale modeling has been widely explored in time-series anomaly detection to capture dynamic patterns over different temporal ranges. MSCRED models inter-variable correlation patterns across multiple temporal spans using multi-scale signature matrices Zhang et al. (2019). DCdetector constructs dual-attention contrastive representations using multiple patch sizes Yang et al. (2023), while CATCH combines frequency patching with channel-aware fusion to enhance multivariate anomaly representations Wu et al. (2025). These methods primarily exploit multi-scale information at the input, feature, or representation level, providing effective strategies for modeling anomalous patterns that manifest over different temporal ranges.

In contrast, this work focuses on the integration of anomaly evidence under scale uncertainty. Anomaly scores produced at different patch scales are not naturally comparable in either their tempo-

ral coverage or numerical ranges. Therefore, rather than directly fusing multi-scale features, MGF-AD independently models normal patterns at each scale, aligns the resulting anomaly responses onto a unified timeline, and performs scale-wise score calibration before integrating them into point-level anomaly evidence. In this way, our focus shifts from multi-scale representation learning toward the alignment, calibration, and fusion of cross-scale anomaly evidence.

B Supplementary Experimental Results

B.1 Subset-Level Anomaly Detection Results

To complement the overall macro-averaged results, Tables 7 and 8 report the complete subset-level anomaly detection performance of MGF-AD on TSB-AD-U and TSB-AD-M, respectively. No point adjustment is applied to any evaluation metric. All results are obtained from three independent runs with random seeds 2025, 2026, and 2027, and are reported as mean ± standard deviation.

Table 7: Subset-level anomaly detection performance of MGF-AD on TSB-AD-U. Results are reported as mean ± standard deviation over three independent runs with random seeds 2025, 2026, and 2027.

Dataset	Name of Subset	# Time Series	Avg. Dim.	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-U	UCR	70	1	0.4282 ± 0.0066	0.9484 ± 0.0005	0.5100 ± 0.0033	0.4379 ± 0.0009	0.9391 ± 0.0004	0.4813 ± 0.0057
	YAHOO	30	1	0.4959 ± 0.0036	0.8242 ± 0.0017	0.3600 ± 0.0049	0.3166 ± 0.0029	0.8183 ± 0.0017	0.3877 ± 0.0026
	WSD	20	1	0.5036 ± 0.0054	0.9413 ± 0.0029	0.5985 ± 0.0053	0.4907 ± 0.0079	0.9390 ± 0.0028	0.5252 ± 0.0065
	CATSv2	1	1	0.3219 ± 0.0061	0.7618 ± 0.0011	0.1956 ± 0.0029	0.4895 ± 0.0022	0.7859 ± 0.0006	0.5363 ± 0.0011
	Daphnet	1	1	0.4419 ± 0.0443	0.9368 ± 0.0104	0.3770 ± 0.0276	0.4613 ± 0.0303	0.9412 ± 0.0073	0.4986 ± 0.0367
	Exathlon	30	1	0.8424 ± 0.0030	0.9682 ± 0.0007	0.6721 ± 0.0114	0.8362 ± 0.0029	0.9671 ± 0.0005	0.8559 ± 0.0014
	IOPS	15	1	0.3166 ± 0.0041	0.9066 ± 0.0019	0.4079 ± 0.0075	0.2166 ± 0.0048	0.8065 ± 0.0029	0.3036 ± 0.0077
	LTDB	8	1	0.6630 ± 0.0043	0.8015 ± 0.0031	0.6773 ± 0.0040	0.6031 ± 0.0038	0.7723 ± 0.0024	0.6303 ± 0.0023
	MGAB	8	1	0.2979 ± 0.0091	0.9822 ± 0.0030	0.4405 ± 0.0012	0.2847 ± 0.0045	0.9819 ± 0.0028	0.3653 ± 0.0050
	MITDB	7	1	0.3816 ± 0.0044	0.9062 ± 0.0030	0.3708 ± 0.0111	0.3253 ± 0.0049	0.8062 ± 0.0041	0.3875 ± 0.0051
	MSL	7	1	0.2416 ± 0.0083	0.7658 ± 0.0023	0.3933 ± 0.0175	0.2139 ± 0.0107	0.7235 ± 0.0026	0.3254 ± 0.0210
	NAB	23	1	0.5439 ± 0.0010	0.8027 ± 0.0045	0.6031 ± 0.0050	0.5326 ± 0.0004	0.7864 ± 0.0056	0.5696 ± 0.0015
	NEK	8	1	0.5032 ± 0.0031	0.7235 ± 0.0064	0.5877 ± 0.0032	0.4879 ± 0.0054	0.7510 ± 0.0073	0.6125 ± 0.0084
	OPPORTUNITY	27	1	0.3112 ± 0.0094	0.7239 ± 0.0063	0.3929 ± 0.0090	0.3209 ± 0.0102	0.7205 ± 0.0058	0.3750 ± 0.0057
	Power	1	1	0.1962 ± 0.0105	0.6806 ± 0.0057	0.2334 ± 0.0082	0.2137 ± 0.0139	0.6609 ± 0.0064	0.2684 ± 0.0065
	SED	2	1	0.9702 ± 0.0011	0.9991 ± 0.0000	0.8628 ± 0.0028	0.8690 ± 0.0053	0.9957 ± 0.0001	0.8467 ± 0.0013
	SMAP	17	1	0.8324 ± 0.0018	0.9327 ± 0.0013	0.7983 ± 0.0038	0.8236 ± 0.0022	0.9302 ± 0.0016	0.7988 ± 0.0031
	SMD	33	1	0.5248 ± 0.0054	0.9249 ± 0.0013	0.5733 ± 0.0030	0.4989 ± 0.0087	0.9261 ± 0.0017	0.5544 ± 0.0039
	Stock	8	1	0.6778 ± 0.0015	0.8078 ± 0.0002	0.1898 ± 0.0165	0.0903 ± 0.0003	0.4844 ± 0.0013	0.1701 ± 0.0000
	SWDB	18	1	0.6942 ± 0.0024	0.9816 ± 0.0001	0.6025 ± 0.0025	0.6360 ± 0.0031	0.9724 ± 0.0001	0.6477 ± 0.0011
	SWaT	1	1	0.0947 ± 0.0004	0.1688 ± 0.0006	0.1358 ± 0.0257	0.0944 ± 0.0005	0.1851 ± 0.0013	0.2153 ± 0.0000
	TAO	2	1	0.7221 ± 0.0022	0.8461 ± 0.0013	0.3210 ± 0.0646	0.1069 ± 0.0026	0.5553 ± 0.0043	0.1874 ± 0.0009
	TODS	13	1	0.7636 ± 0.0029	0.8889 ± 0.0008	0.3357 ± 0.0085	0.3820 ± 0.0030	0.7836 ± 0.0017	0.4233 ± 0.0038
	Average	–	–	$\mathbf{0.5435 \pm 0.0016}$	$\mathbf{0.8952 \pm 0.0014}$	$\mathbf{0.5084 \pm 0.0006}$	$\mathbf{0.4836 \pm 0.0016}$	$\mathbf{0.8774 \pm 0.0021}$	$\mathbf{0.5332 \pm 0.0017}$

Table 8: Subset-level anomaly detection performance of MGF-AD on TSB-AD-M. Results are reported as mean ± standard deviation over three independent runs with random seeds 2025, 2026, and 2027.

Dataset	Name of Subset	# Time Series	Avg. Dim.	VUS-PR	VUS-ROC	Range-F1	AUC-PR	AUC-ROC	Point-F1
TSB-AD-M	CATSv2	5	17	0.0946 ± 0.0034	0.7052 ± 0.0030	0.1579 ± 0.0094	0.1045 ± 0.0029	0.6984 ± 0.0039	0.1590 ± 0.0064
	CreditCard	1	29	0.0232 ± 0.0052	0.5156 ± 0.0381	0.0162 ± 0.0084	0.0032 ± 0.0015	0.5192 ± 0.0383	0.0265 ± 0.0183
	Daphnet	1	9	0.2488 ± 0.0083	0.8810 ± 0.0041	0.2697 ± 0.0161	0.2270 ± 0.0068	0.8997 ± 0.0043	0.3851 ± 0.0059
	Exathlon	25	20.16	0.8053 ± 0.0038	0.9790 ± 0.0023	0.6970 ± 0.0046	0.7730 ± 0.0032	0.9765 ± 0.0023	0.8753 ± 0.0021
	GECCO	1	9	0.2202 ± 0.0199	0.9216 ± 0.0087	0.3028 ± 0.0223	0.2603 ± 0.0147	0.9388 ± 0.0049	0.3304 ± 0.0033
	GHL	23	19	0.0094 ± 0.0000	0.3519 ± 0.0010	0.0258 ± 0.0013	0.0081 ± 0.0000	0.3211 ± 0.0011	0.0262 ± 0.0001
	Genesis	1	18	0.2113 ± 0.0040	0.9900 ± 0.0007	0.3673 ± 0.0367	0.1134 ± 0.0083	0.9888 ± 0.0008	0.2638 ± 0.0193
	LTDB	4	2.25	0.6809 ± 0.0239	0.8944 ± 0.0062	0.6885 ± 0.0332	0.6608 ± 0.0207	0.8671 ± 0.0060	0.6488 ± 0.0155
	MITDB	11	2	0.4005 ± 0.0058	0.8983 ± 0.0013	0.4925 ± 0.0120	0.4564 ± 0.0073	0.8757 ± 0.0013	0.4977 ± 0.0065
	MSL	14	55	0.2835 ± 0.0383	0.8098 ± 0.0088	0.4290 ± 0.0098	0.2486 ± 0.0476	0.7930 ± 0.0078	0.3550 ± 0.0286
	OPPORTUNITY	7	248	0.1654 ± 0.0160	0.5652 ± 0.0244	0.3450 ± 0.0362	0.1567 ± 0.0146	0.5306 ± 0.0229	0.2013 ± 0.0146
	PSM	1	25	0.2098 ± 0.0089	0.6520 ± 0.0077	0.3027 ± 0.0059	0.1925 ± 0.0095	0.6570 ± 0.0145	0.2603 ± 0.0025
	SMAP	25	25	0.5163 ± 0.0042	0.8936 ± 0.0030	0.5570 ± 0.0078	0.4915 ± 0.0060	0.8991 ± 0.0042	0.5172 ± 0.0075
	SMD	20	38	0.4019 ± 0.0079	0.8657 ± 0.0052	0.4389 ± 0.0084	0.4171 ± 0.0059	0.8833 ± 0.0032	0.4683 ± 0.0081
	SVDB	28	2	0.5679 ± 0.0058	0.9108 ± 0.0008	0.5808 ± 0.0084	0.5467 ± 0.0073	0.8875 ± 0.0012	0.5610 ± 0.0072
	SWaT	2	58.5	0.2414 ± 0.0082	0.6838 ± 0.0216	0.3669 ± 0.0317	0.2107 ± 0.0077	0.6620 ± 0.0244	0.3239 ± 0.0094
	TAO	11	3	0.7470 ± 0.0025	0.8695 ± 0.0008	0.2410 ± 0.0299	0.0857 ± 0.0013	0.4954 ± 0.0053	0.1628 ± 0.0005
	Average	–	–	$\mathbf{0.4425 \pm 0.0034}$	$\mathbf{0.8060 \pm 0.0008}$	$\mathbf{0.4399 \pm 0.0010}$	$\mathbf{0.3895 \pm 0.0033}$	$\mathbf{0.7715 \pm 0.0009}$	$\mathbf{0.4396 \pm 0.0031}$

As shown in Tables 7 and 8, the detection performance of MGF-AD varies substantially across different subsets. This variation indicates that overall macro-averaged results alone cannot fully characterize model behavior across heterogeneous time-series sources. We therefore additionally report subset-level results to provide a more fine-grained view of performance across different datasets.

B.2 Supplementary Analysis of Multi-Scale Responses and Scale Disagreement

Multi-scale response complementarity. Different patch lengths correspond to different temporal receptive fields. Consequently, even for the same anomalous event, the point-wise anomaly responses produced at different scales need not be identical. Figure 4 illustrates this behavior using the `518_UCR` example, showing the responses of Scale-64, Scale-96, Scale-128, and their fused result. For each scale, the patch-level anomaly scores are first mapped back onto the original timeline and then processed using the same per-scale calibration procedure as in the main method, ensuring that the responses are comparable across scales.

Around the ground-truth anomalous interval near $t = 21{,}000$, all three scales produce clear anomaly responses, while their peak magnitudes and temporal coverage differ. Meanwhile, in the normal region around $t = 21{,}400$, Scale-64 exhibits a strong isolated response, whereas Scale-96, Scale-128, and the fused result remain at relatively low levels. In this example, the anomaly responses from different scales therefore do not completely overlap, and multi-scale fusion attenuates an isolated high response that appears only at a single scale.

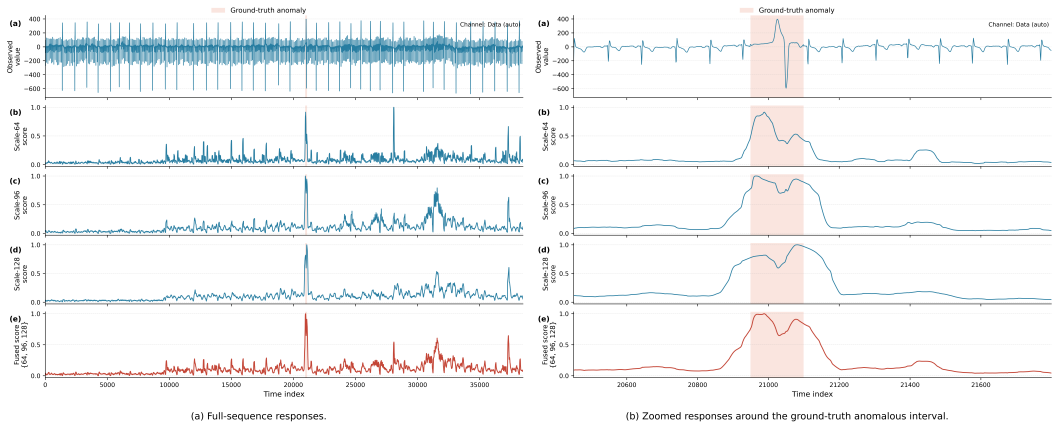


Figure 4: Multi-scale anomaly responses on the `518_UCR` example. Calibrated point-wise responses from Scale-64, Scale-96, and Scale-128 are shown together with the fused anomaly score, illustrating shared and scale-specific responses.

Scale-disagreement analysis. To quantitatively characterize the differences among scale-specific outputs in a representative example, we define scale disagreement as the standard deviation of the calibrated anomaly scores across scales at each timestamp t. Specifically, the mean cross-scale response is

$$\bar{s}_t = \frac{1}{M} \sum_{m=1}^{M} \tilde{s}_t^{(m)}, \tag{22}$$

and the corresponding scale disagreement is defined as

$$D_t = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\tilde{s}_t^{(m)} - \bar{s}_t \right)^2}. \tag{23}$$

A larger D_t indicates that different temporal receptive fields produce less consistent anomaly assessments at timestamp t.

On the `145_MSL` example shown in Figure 5, the mean scale disagreement within the ground-truth anomalous region is 0.3474, compared with 0.1082 in the normal region, making the former approximately $3.21\times$ larger than the latter. Moreover, when D_t alone is treated as a point-wise score for distinguishing anomalous from normal timestamps in this sequence, it achieves a ROC-AUC of 0.9634. These observations indicate that, in this particular example, the anomalous region is accompanied by substantially stronger cross-scale response disagreement.

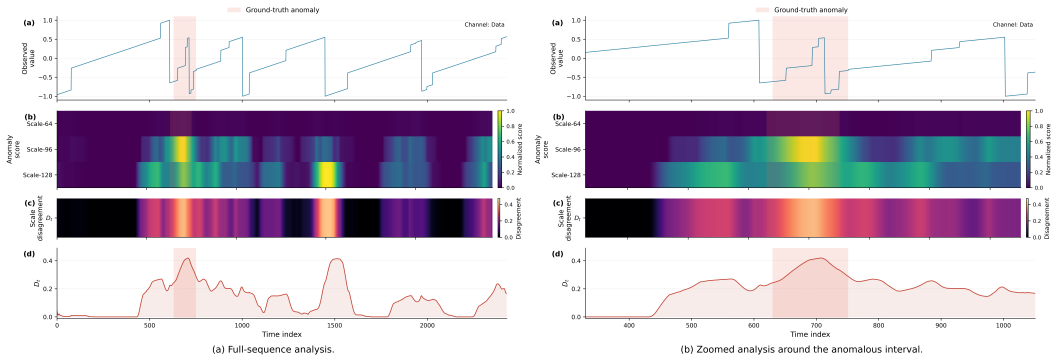


Figure 5: Scale-disagreement analysis on the `145_MSL` example. The figure shows scale-specific calibrated responses and the resulting disagreement score D_t. The anomalous interval exhibits stronger cross-scale disagreement than the surrounding normal regions.

Figures 4 and 5 provide two concrete examples illustrating differences among multi-scale anomaly responses. They show that different temporal receptive fields can produce both jointly elevated anomaly responses and scale-specific responses. Since these examples are intended as qualitative, mechanism-level illustrations rather than general statistical evidence, the broader utility of temporal-scale diversity is evaluated primarily through the homogeneous ensemble control and anomaly-duration sensitivity analysis presented in the main text.

B.3 Complete Results of Best-Single Oracle-Gap Analysis

Best-single oracle-gap analysis. For an evaluation metric m, let $m_{i,s}$ denote the performance of the i-th test sequence under a fixed patch scale $s \in \{64, 96, 128\}$. The Best Fixed baseline selects a single scale that achieves the highest macro-averaged performance over the entire Eval split:

$$B = \max_s \frac{1}{N} \sum_{i=1}^{N} m_{i,s}, \tag{24}$$

where N denotes the number of test sequences.

In contrast, the Best-single Oracle retrospectively selects, for each test sequence, the fixed scale that yields the best performance:

$$O = \frac{1}{N} \sum_{i=1}^{N} \max_s m_{i,s}. \tag{25}$$

Let F denote the performance achieved by multi-scale Fusion. We define the fraction of the oracle gap recovered by Fusion as

$$R_{\text{gap}} = \frac{F - B}{O - B} \times 100\%. \tag{26}$$

The Oracle uses ground-truth labels for retrospective per-sequence scale selection and therefore serves only as a diagnostic performance upper bound rather than a deployable model-selection strategy. Complete Range-F1 and VUS-PR results on both benchmarks are reported in Table 9.

As shown in Table 9, a clear performance gap exists between the globally selected Best Fixed scale and the per-series Best-single Oracle on both benchmarks. This indicates that a single dataset-level patch scale cannot simultaneously match the preferred temporal granularity of all test sequences.

Without using test labels for scale selection, multi-scale Fusion partially narrows this gap. Specifically, it recovers 31.66% and 40.46% of the oracle gap on TSB-AD-U in terms of Range-F1 and VUS-PR, respectively, and recovers 26.35% and 11.81% on TSB-AD-M. These results further support the motivation for multi-scale evidence fusion under scale uncertainty.

Table 9: Best-single oracle-gap analysis on the Eval splits of TSB-AD-U and TSB-AD-M. B, F, and O denote the performance of the best fixed scale, multi-scale Fusion, and per-series Best-single Oracle, respectively. Gap Recovery is the fraction of the B–O gap recovered by Fusion.

Dataset	Metric	Best Fixed Scale	B	Fusion F	Oracle O	Gain $(F - B)$	Gap Recovery (%)
TSB-AD-U	Range-F1	Scale-64	0.4622	0.5077	0.6059	0.0455	31.66
	VUS-PR	Scale-96	0.4866	0.5417	0.6228	0.0551	40.46
TSB-AD-M	Range-F1	Scale-96	0.4261	0.4412	0.4834	0.0151	26.35
	VUS-PR	Scale-96	0.4389	0.4451	0.4914	0.0062	11.81

B.4 Computational Cost

Runtime and computational cost. Multi-scale fusion requires independently executing multiple scale-specific branches. We therefore further measure the average end-to-end runtime per test sequence on the Eval split.

For a configuration c, we define its average runtime as

$$T_c = \frac{1}{N} \sum_{i=1}^{N} T_{c,i}, \tag{27}$$

where $T_{c,i}$ denotes the end-to-end inference time of configuration c on the i-th test sequence, and N is the number of sequences in the corresponding Eval split.

To quantify relative computational cost, we use the fastest fixed-scale configuration on the same benchmark as the reference. Specifically,

$$R_c = \frac{T_c}{\min_{s \in \{64,96,128\}} T_s}, \tag{28}$$

where R_c denotes the runtime cost of configuration c relative to the fastest fixed-scale variant.

Table 10 reports the average per-sequence end-to-end runtime and relative computational cost of the fixed-scale variants and multi-scale Fusion on the Eval splits. All runtime measurements are obtained on a single NVIDIA A800 80GB GPU.

Table 10: Average per-sequence end-to-end runtime and relative computational cost on the Eval splits. Relative cost is computed with respect to the fastest fixed-scale configuration on each benchmark.

Dataset	Configuration	Mean Time/File (s)	Relative Cost
TSB-AD-U	Scale-64	4.929	1.00×
	Scale-96	4.928	1.00×
	Scale-128	5.240	1.06×
	Fusion	14.633	2.97×
TSB-AD-M	Scale-64	10.169	1.00×
	Scale-96	10.644	1.05×
	Scale-128	11.099	1.09×
	Fusion	32.138	3.16×

As shown in Table 10, under the current serial implementation, the runtime of Fusion on TSB-AD-U and TSB-AD-M is approximately $2.97\times$ and $3.16\times$ that of the fastest fixed-scale configuration,

respectively. This computational overhead is broadly consistent with an architecture composed of three independently executed scale-specific branches.

Importantly, interactions among scales occur only at the final point-wise evidence-fusion stage. The scale-specific branches are therefore structurally independent and naturally amenable to parallel execution. Consequently, the runtime reported here reflects the current serial implementation rather than an inherent latency requirement of the multi-scale design.

C Implementation and Reproducibility Details

C.1 Model Architecture and Training Settings

MGF-AD uses three patch scales $\{64, 96, 128\}$ by default, with the sliding stride set to 1. The three scales independently construct their own patch encoders and Normal Prototype Banks, while all other training hyperparameters are kept identical across scales. Each input patch is normalized using non-affine RevIN before being fed into the encoder.

For each scale, the patch encoder is implemented as a four-layer 1D-CNN. The numbers of convolutional channels are

$$[128, 256, 128, 64], \tag{29}$$

and the corresponding kernel sizes are

$$[7, 5, 3, 3]. \tag{30}$$

The resulting patch embedding has a dimensionality of 64. The projection head is implemented as a two-layer MLP with both hidden and output dimensions set to 256, while the temporal-continuity classification head is implemented as a single-layer MLP.

Each scale is trained independently for 100 iterations with a mini-batch size of 512. We use AdamW as the optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. A cosine-annealing schedule gradually decreases the learning rate to 1×10^{-5}. For neighborhood-consistency metric learning, the positive-patch radius is set to $r = 2$, and the triplet margin is set to $\delta = 0.1$. For the temporal-continuity auxiliary task, $U = 5$ non-contiguous patches are randomly sampled for each anchor. The auxiliary loss is active during the first 20% of the training iterations, with its weight linearly decaying to zero over this period.

After representation learning, a Normal Prototype Bank is independently constructed for each scale using MiniBatch k-means over L_2-normalized normal training embeddings. For each cluster, the training embedding nearest to the centroid is retained as its representative prototype. The bank size is set to 10% of the normal training patch collection.

The complete model architecture, optimization settings, prototype-bank configuration, and default cross-scale evidence-processing hyperparameters are summarized in Table 11.

Table 11: Implementation details and hyperparameter settings of MGF-AD.

Setting	Value	Setting	Value	Setting	Value
Patch scales	$\{64, 96, 128\}$	Normalization	RevIN, non-affine	Positive-patch radius r	2
Patch stride	1	Fusion method	Arithmetic mean	Triplet margin δ	0.1
Scale-specific modeling	Independent encoder and prototype bank for each scale	Score calibration	Per-scale min–max normalization	Random patches per anchor U	5
Patch encoder	Four-layer 1D-CNN	Training iterations	100 per scale	Auxiliary-loss schedule	Linear decay during the first 20% of iterations
CNN channels	$[128, 256, 128, 64]$	Batch size	512	Prototype-bank ratio	10%
Kernel sizes	$[7, 5, 3, 3]$	Optimizer	AdamW	Number of nearest neighbors	$k = 1$
Patch embedding dimension	64	Initial learning rate	1×10^{-4}	Random seed	2027
Projection head	Two-layer MLP with hidden/output dimension 256	Learning-rate schedule	Cosine annealing to 1×10^{-5}	Hardware	NVIDIA A800 80GB
Classification head	One-layer MLP	Weight decay	1×10^{-4}	Software	Python 3.8.20, PyTorch 2.0.1

C.2 Baselines and Internal Controls

Baselines. To comprehensively evaluate MGF-AD, we compare it against representative methods spanning statistical and machine-learning approaches, conventional neural networks, Transformer-based architectures, and time-series foundation models.

The statistical and machine-learning baselines include KMeansAD, (Sub)-PCA Aggarwal (2016), KShapeAD Paparrizos & Gravano (2017), DLinear, and NLinear Zeng et al. (2023). Conventional neural-network-based methods include DeepAnT Munir et al. (2018), USAD Audibert et al. (2020), OmniAnomaly Su et al. (2019), TimesNet Wu et al. (2023), FITS Xu et al. (2024), DADA Shentu et al. (2025), and KAN-AD Zhou et al. (2025). Transformer-based and foundation-model baselines include Anomaly Transformer Xu et al. (2021), DCdetector Yang et al. (2023), Lag-Llama Rasul et al. (2023), OFA Zhou et al. (2023), PatchTST Nie et al. (2023), iTransformer Liu et al. (2024), CATCH Wu et al. (2025), MOMENT Goswami et al. (2024), and TimesFM Das et al. (2024). In addition, we place particular emphasis on PaAno Park & Kang (2026), a patch-based anomaly detection method that is closely related to MGF-AD.

Together, these baselines cover a broad range of representative technical paradigms, including classical statistical modeling, reconstruction-based anomaly detection, attention-based architectures, and pretrained time-series foundation models.

Internal fixed-scale controls. To further isolate the contribution of the proposed multi-scale mechanism, we construct three internal fixed-scale baselines, denoted as Scale-64, Scale-96, and Scale-128. These variants use the same encoder architecture, training objectives, Normal Prototype Bank construction, and anomaly-scoring pipeline as MGF-AD. The only difference is that each variant retains a single patch scale rather than combining multiple temporal granularities. These controlled baselines therefore enable us to quantify the performance gains specifically attributable to multi-scale anomaly evidence fusion.

C.3 Randomness and Repeated Experiments

Unless otherwise specified, all single-run experiments of MGF-AD and its internal control variants use random seed 2027. For experiments whose results are reported as mean $\pm$ standard deviation, we conduct three independent training and evaluation runs using random seeds 2025, 2026, and 2027, and report the mean and standard deviation across the three runs.

Specifically, any result explicitly reported with "$\pm$" in a table or figure corresponds to the three-seed repeated experiments described above, whereas results without "$\pm$" correspond to a single run with random seed 2027. Across different random seeds, the dataset splits, model architecture, training hyperparameters, and evaluation protocol are kept identical; only the random initialization and states associated with stochastic sampling are changed.

The multi-scale versus homogeneous ensemble control is likewise repeated three times using random seeds 2025, 2026, and 2027. This design reduces the influence of a single random initialization on the comparison between homogeneous and heterogeneous scale configurations.

D Evaluation Metrics and Aggregation Protocol

D.1 Evaluation Protocol

We follow the predefined Tuning/Eval splits of TSB-AD-U and TSB-AD-M. The Tuning split is used exclusively for model configuration and hyperparameter selection, whereas the Eval split is reserved for final performance reporting.

We report six evaluation metrics: VUS-PR, VUS-ROC, Range-F1, AUC-PR, AUC-ROC, and Point-F1. Among them, VUS-PR, VUS-ROC, and AUC-based metrics do not rely on a single predefined detection threshold, while Range-F1 and Point-F1 complement them by evaluating detection quality at the anomalous-segment and timestamp levels, respectively. No point adjustment is applied to any metric. We use VUS-PR as the primary evaluation metric.

Consider a test sequence of length T, with ground-truth anomaly labels

$$y_t \in \{0, 1\}, \qquad t = 1, \ldots, T, \tag{31}$$

where $y_t = 1$ indicates an anomalous timestamp and $y_t = 0$ indicates a normal timestamp. Let s_t denote the continuous anomaly score produced by the model. Given a threshold τ, the corresponding binary prediction is

$$\hat{y}_t(\tau) = \mathbb{I}\left[s_t \geq \tau\right], \tag{32}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function.

AUC-ROC. For a given threshold τ, the true-positive rate and false-positive rate are defined as

$$\mathrm{TPR}(\tau) = \frac{\sum_{t=1}^{T} \hat{y}_t(\tau) y_t}{\sum_{t=1}^{T} y_t}, \tag{33}$$

and

$$\mathrm{FPR}(\tau) = \frac{\sum_{t=1}^{T} \hat{y}_t(\tau)(1 - y_t)}{\sum_{t=1}^{T} (1 - y_t)}. \tag{34}$$

Varying τ yields the receiver operating characteristic (ROC) curve, and AUC-ROC is the area under this curve. It measures the overall ranking ability of the anomaly scores in distinguishing anomalous from normal timestamps.

AUC-PR. For a given threshold τ, precision and recall are defined as

$$P(\tau) = \frac{\sum_{t=1}^{T} \hat{y}_t(\tau) y_t}{\sum_{t=1}^{T} \hat{y}_t(\tau)}, \tag{35}$$

and

$$R(\tau) = \frac{\sum_{t=1}^{T} \hat{y}_t(\tau) y_t}{\sum_{t=1}^{T} y_t}. \tag{36}$$

Varying τ produces the precision–recall (PR) curve, and AUC-PR is the area under this curve. Since time-series anomaly detection typically involves severe class imbalance, AUC-PR provides a useful measure of the model's ability to distinguish the minority anomalous class.

Point-F1. Given threshold τ, the point-wise F1 score is

$$F_1(\tau) = \frac{2P(\tau)R(\tau)}{P(\tau) + R(\tau)}. \tag{37}$$

Point-F1 is defined as the maximum F1 score over all candidate thresholds:

$$\text{Point-F1} = \max_{\tau} F_1(\tau). \tag{38}$$

Point-F1 directly evaluates timestamp-level predictions, and no point adjustment is applied during its computation.

Range-F1. In addition to timestamp-level classification accuracy, time-series anomaly detection also requires evaluating whether contiguous anomalous events are successfully detected. Let the set of ground-truth anomalous intervals be

$$\mathcal{A} = \{A_1, A_2, \ldots, A_K\}, \tag{39}$$

and let the predicted anomalous intervals obtained under threshold τ be

$$\hat{\mathcal{A}}(\tau) = \left\{\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_{\hat{K}}\right\}. \tag{40}$$

A predicted anomalous interval is considered matched if it overlaps with at least one ground-truth anomalous interval. Range precision is therefore defined as

$$\text{Range-P}(\tau) = \frac{\left|\left\{\hat{A} \in \hat{\mathcal{A}}(\tau) : \exists A \in \mathcal{A},\ \hat{A} \cap A \neq \varnothing\right\}\right|}{\left|\hat{\mathcal{A}}(\tau)\right|}. \tag{41}$$

Similarly, a ground-truth anomalous interval is considered detected if it overlaps with at least one predicted anomalous interval. Range recall is defined as

$$\text{Range-R}(\tau) = \frac{\left|\left\{A \in \mathcal{A} : \exists \hat{A} \in \hat{\mathcal{A}}(\tau),\ A \cap \hat{A} \neq \varnothing\right\}\right|}{|\mathcal{A}|}. \tag{42}$$

The final Range-F1 score is

$$\text{Range-F1} = \max_{\tau} \frac{2\,\text{Range-P}(\tau)\text{Range-R}(\tau)}{\text{Range-P}(\tau) + \text{Range-R}(\tau)}. \tag{43}$$

Compared with Point-F1, this metric evaluates detection quality at the anomalous-event level and therefore emphasizes whether anomalous intervals are successfully identified rather than only whether individual timestamps are correctly classified.

VUS-ROC and VUS-PR. The volume-under-the-surface (VUS) metrics extend conventional ROC- and PR-based evaluation by introducing temporal tolerance into the assessment of anomalous intervals. Instead of evaluating predictions under a single exact temporal alignment, VUS considers a range of temporal tolerance widths around anomalous regions.

VUS-ROC jointly varies the detection threshold and temporal tolerance range to construct an ROC surface and computes the volume under that surface. VUS-PR follows the same principle using precision–recall surfaces. Compared with standard AUC-ROC and AUC-PR, VUS metrics evaluate not only the ranking quality of anomaly scores but also the temporal localization relationship between detected anomalies and ground-truth anomalous intervals. In our experiments, we use the VUS evaluation protocol provided by TSB-AD.

D.2 File-Level Macro Averaging

To prevent long time series or large subsets from receiving disproportionate weight in aggregate results, we first compute each evaluation metric independently for every test time series and then perform file-level macro averaging over all sequences in the Eval split.

For an evaluation metric $\mathcal{M}$, the macro-averaged result is defined as

$$\mathcal{M}_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{M}_i, \tag{44}$$

where $\mathcal{M}_i$ denotes the metric value computed on the i-th test time series, and N is the total number of test sequences included in the evaluation.