WAVELET-GUIDED BRANCH-CONDITIONED GROUPED NORMALIZING FLOW FOR POWER-LINE ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

UAV power-line inspection requires anomaly detectors to distinguish equipment faults from substantial changes in background, viewpoint, and illumination. We propose Wavelet-Guided Branch-Conditioned Grouped Normalizing Flow (WBGFlow), a shared conditional density model that organizes complementary observations for this setting. RAW appearance, low-frequency structure, and high-frequency energy retain distinct feature groups throughout the flow. Per-block identity embeddings adapt shared coupling subnetworks to each observation source. All three branches learn normality jointly, while the final score selects RAW and LL latent energies. This connects complementary-observation training with explicit control over anomaly-ranking evidence. On InsPLAD-fault, WBGFlow achieves the highest mean image-level AUROC among the compared methods at 94.62%, averaged over two training seeds, exceeding DifferNet by 2.16 percentage points in macro-average with gains in all five evaluated categories. A complementary MVTec AD evaluation yields 95.78%. Component ablations examine the roles of grouping and identity conditioning, while fixed-model scoring analysis supports separating training observations from ranking contributions. Together, the results demonstrate the value of identity-preserving shared density modeling for uncontrolled power-line inspection.

1 INTRODUCTION

Reliable anomaly detection in UAV power-line inspection requires separating equipment faults from changes in image acquisition. Background, viewpoint, illumination, and scale vary substantially, while faults include missing insulator units, corrosion, and attached bird nests. InsPLAD captures this combination in real inspection imagery (Vieira e Silva et al., 2023); CableInspect-AD documents related acquisition challenges in robotic inspection (Arodi et al., 2024). Fault examples are limited and cover different types of damage. Learning from normal images is useful in this setting (Ruff et al., 2018). The anomaly score needs to distinguish equipment faults from changes in imaging conditions.

PatchCore detects anomalies through distances to stored normal features (Roth et al., 2022). Other methods use teacher-student differences (Deng & Li, 2022) or feature reconstruction errors (Guo et al., 2025). These scores can respond to changes in both equipment and background. DifferNet fits a normalizing flow to global descriptors from several image scales (Rudolph et al., 2021). It models normal features without a patch memory bank. Flow models may still rely on local image statistics (Kirichenko et al., 2020), and high likelihood does not always indicate normal content (Nalisnick et al., 2019). We therefore examine which observations enter the flow and how their outputs form the anomaly score.

We build on DifferNet and organize its inputs by frequency. Rescaling an image changes both object size and visible detail. Frequency separation gives the branches more specific roles. RAW retains the original appearance. LL represents locally averaged structure. HF contains directional detail energy. Fine edges and surface details in normal images can help the model learn equipment appearance. However, HF also responds to wires, vegetation, and imaging noise. A branch that helps training may therefore be less useful when its score is added at test time.

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.

Within each flow block, RAW, LL, and HF share the coupling subnetworks. Feature permutations and coupling operations remain within each branch. Each block also learns an identity embedding for each branch. The shared subnetworks use this embedding to predict branch-dependent scales and shifts. During training, all three branch losses update the shared weights. At test time, we average the RAW and LL scores. HF does not enter this average, but its training loss affects the weights used by both scoring branches.

Our contributions are as follows:

1. We propose **wavelet-guided shared conditional density modeling** for uncontrolled inspection. Frequency observations, grouped coupling, and per-block identity embeddings preserve observation identity while allowing branches to learn shared transformations.

2. We introduce **distinct training and scoring roles within the shared model**. RAW, LL, and HF jointly train the flow, while branch-preserving transformations make RAW+LL scoring possible without adding the direct HF score.

3. WBGFlow achieves a **two-seed mean AUROC of 94.62%** on InsPLAD-fault, improving on DifferNet by **2.16 percentage points** with gains in all five categories. MVTec AD provides a complementary evaluation at 95.78%. Component and scoring analyses examine how the connected design choices contribute to these results.

2 Related Work

Industrial visual anomaly detection. PaDiM fits Gaussian distributions to local features (Defard et al., 2021). PatchCore compares test patches with stored normal patches (Roth et al., 2022). RD4AD uses reverse distillation (Deng & Li, 2022), which RD++ further develops (Tien et al., 2023). EfficientAD combines teacher-student prediction with an autoencoder (Batzner et al., 2024). CutPaste creates training examples through image perturbations (Li et al., 2021). DRAEM combines reconstruction and discrimination (Zavrtanik et al., 2021), while SimpleNet creates pseudo-anomalies in feature space (Liu et al., 2023). For multi-class detection, UniAD reconstructs normal features (You et al., 2022). Dinomaly restricts feature reconstruction (Guo et al., 2025), and INP-Former uses prototypes obtained from individual images (Luo et al., 2025). CLIP provides image-text representations (Radford et al., 2021), while DINOv2 provides self-supervised visual features (Oquab et al., 2024). WinCLIP uses prompts and image windows (Jeong et al., 2023). AnomalyCLIP learns object-agnostic prompts (Zhou et al., 2024). MuSc scores unlabeled images through comparisons with other images (Li et al., 2024).

Normalizing flows and frequency representations. Real NVP estimates densities through affine coupling layers (Dinh et al., 2017). Glow adds invertible channel mixing (Kingma & Dhariwal, 2018). DifferNet models global descriptors from several scales (Rudolph et al., 2021). CS-Flow models relations across scales (Rudolph et al., 2022), while CFLOW-AD conditions its flow on spatial positions (Gudovskiy et al., 2022). FastFlow applies two-dimensional flows to feature maps (Yu et al., 2021). MSFlow combines density information from multiple scales (Zhou et al., 2025). Local image statistics can influence flow-based anomaly scores (Kirichenko et al., 2020). Several methods also use frequency information. Wavelet Flow models wavelet coefficients across scales (Yu et al., 2020). FcaNet uses frequency components for channel attention (Qin et al., 2021). WE-PaDiM combines wavelet subbands with local Gaussian models (Gardner et al., 2026).

Inspection in uncontrolled environments. MVTec AD contains industrial objects and textures captured under controlled conditions (Bergmann et al., 2019). MVTec AD 2 adds illumination changes and more challenging defects (Heckler-Kram et al., 2026). InsPLAD provides UAV images of power-line equipment with fault annotations (Vieira e Silva et al., 2023). CableInspect-AD covers complex conditions in robotic inspection (Arodi et al., 2024). AttentDifferNet introduces attention into power-line anomaly detection (Vieira e Silva et al., 2024). Its variants use SE channel attention (Hu et al., 2018) or CBAM channel and spatial attention (Woo et al., 2018).

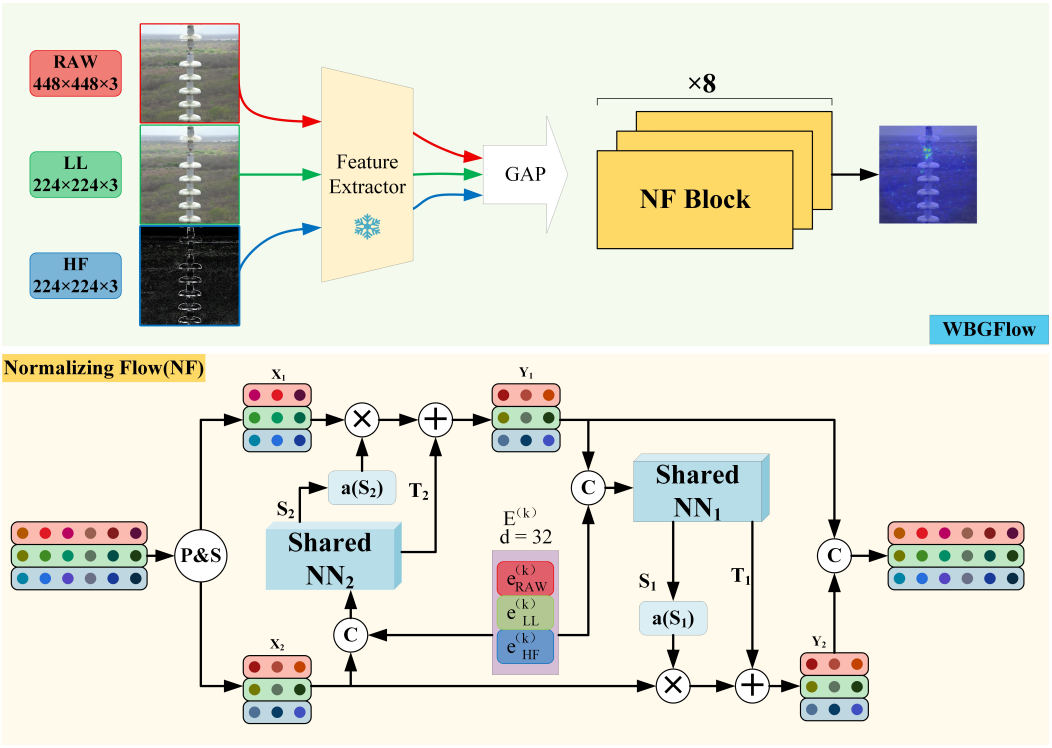

Figure 1: WBGFlow architecture and grouped coupling block. The three branches share a frozen AlexNet and eight flow blocks. Each block contains two subnetworks with separate weights. Each subnetwork shares its weights across branches and receives a 32-dimensional branch embedding. P&S denotes within-branch permutation and splitting. C denotes concatenation. All branches contribute to training. The final score uses RAW and LL. Equation (E.2) defines the displayed saliency map.

3 METHOD

3.1 OVERALL FRAMEWORK

For each asset category, WBGFlow learns from normal images and ranks normal and anomalous test images by a scalar score. Figure 1 shows the architecture. RAW, LL, and HF share the frozen convolutional feature extractor of an ImageNet-pretrained AlexNet (Krizhevsky et al., 2012). Global average pooling produces one 256-dimensional descriptor per branch, giving a 768-dimensional flow state. Eight branch-conditioned grouped coupling blocks map these descriptors into a standard Gaussian latent space. All branches train the shared flow, while inference fuses RAW and LL latent energies. Formal dataset and feature definitions are provided in Appendix A.

The architecture separates shared representation from branch-specific transformation. The frozen encoder places all observations in a common feature space. Within that space, grouping preserves the origin of each descriptor, and identity conditions adapt shared coupling weights to the statistics of each branch. Joint training connects the branch objectives through those weights. Separate latent energies then make the scoring rule explicit, so each design choice supports the next stage of the model.

3.2 WAVELET-GUIDED OBSERVATIONS

The Haar transform separates approximation and directional detail using fixed local operations (Mallat, 1989). Let $x_{00}, x_{01}, x_{10}, x_{11}$ denote the four pixel positions sampled from each two-by-two neighborhood. The low-frequency observation uses local averaging:

$$x_{\mathrm{LL}} = \frac{x_{00} + x_{01} + x_{10} + x_{11}}{4}. \tag{1}$$

We square and average the directional detail components to prevent signed responses from canceling:

$$x_{\mathrm{HF}} = \frac{x_{\mathrm{LH}}^2 + x_{\mathrm{HL}}^2 + x_{\mathrm{HH}}^2}{3}. \tag{2}$$

RAW is the 448×448 RGB input. LL and HF each have a resolution of 224×224. Both are computed separately for each color channel. LL smooths local variation. HF contains detail responses from equipment and background. The flow keeps these observations in separate feature groups. Appendix A.2 gives the sampling indices and directional filters.

3.3 Branch-Conditioned Grouped Coupling

Each coupling block permutes features only within their 256-dimensional branch and splits them into two 128-dimensional halves, $u_{b,1}$ and $u_{b,2}$. A learnable 32-dimensional identity vector $e_b^{(k)}$ then conditions the shared subnetworks. Thus permutation and splitting precede identity concatenation. The embeddings enter the parameter predictors, not the invertible state; its dimension remains 768. Grouping preserves the source of each feature, while identity conditioning lets shared weights adapt the transformation to that source.

Omitting block index k, the first affine step uses the second half and its identity to update the first:

$$\begin{aligned} (s_{b,2}, t_{b,2}) &= \mathrm{split}\big(g_{\theta,2}([u_{b,2}; e_b])\big), \\ y_{b,1} &= \exp\big(C_\alpha(s_{b,2})\big) \odot u_{b,1} + t_{b,2}. \end{aligned} \tag{3}$$

The second step uses the updated first half to transform the second:

$$\begin{aligned} (s_{b,1}, t_{b,1}) &= \mathrm{split}\big(g_{\theta,1}([y_{b,1}; e_b])\big), \\ y_{b,2} &= \exp\big(C_\alpha(s_{b,1})\big) \odot u_{b,2} + t_{b,1}. \end{aligned} \tag{4}$$

Here, $[\cdot;\cdot]$ denotes concatenation, $\odot$ denotes elementwise multiplication, and split divides a 256-dimensional output into log-scale and shift vectors. The arctangent soft clamp C_α uses $\alpha = 3$. Each subnetwork shares weights across RAW, LL, and HF within a block; the two subnetworks and different blocks have separate parameters. For fixed parameters, each output depends only on its own branch input, giving a block-diagonal feature Jacobian. Appendix A.4 provides the permutation formula, dimensions, and figure notation.

Grouping and identity conditioning have complementary roles. Grouping retains the observation associated with each latent coordinate, while conditioning gives shared parameter predictors access to that observation identity. The grouped Jacobian describes this structural separation; observations from the same image can still be statistically related. AttentDifferNet recalibrates convolutional features with image-dependent attention (Vieira e Silva et al., 2024). WBGFlow instead conditions the density transformation on a fixed branch identity, preserving the original state dimension while adapting the affine parameters to each source.

3.4 Shared Maximum-Likelihood Training

Let $\mathcal{L}_b$ be branch b's dimension-normalized negative log-likelihood under a standard Gaussian prior, and θ the weights shared across branches. With $D_b = 256$ and $D = 768$, group-preserving transformations allow the total energy and log-determinant to be summed across branches:

$$\begin{aligned} \mathcal{L}_{\mathrm{NF}} &= \sum_{b \in \{\mathrm{RAW,LL,HF}\}} \frac{D_b}{D} \mathcal{L}_b = \frac{1}{3} \sum_b \mathcal{L}_b, \\ \nabla_\theta \mathcal{L}_{\mathrm{NF}} &= \frac{1}{3} \left(\nabla_\theta \mathcal{L}_{\mathrm{RAW}} + \nabla_\theta \mathcal{L}_{\mathrm{LL}} + \nabla_\theta \mathcal{L}_{\mathrm{HF}} \right). \end{aligned} \tag{5}$$

The likelihood models the frozen encoder's feature space, with both latent energy and the log-determinant contributing to training. Equation (5) makes the connection between HF observations and final scoring explicit: the HF loss updates weights also used to transform RAW and LL. At inference, each branch is evaluated from its own descriptor and identity. HF therefore influences RAW+LL scoring through learned shared parameters rather than an additional test-time feature

input. Appendix A.5 derives the feature-space objective, and Appendix A.6 specifies its numerical implementation and density interpretation.

3.5 Selective Anomaly Scoring

For $T = 64$ transformed views $\tau_t(x)$, each branch score averages squared latent energy over views and its $D_b = 256$ dimensions:

$$S_b(x) = \frac{1}{T}\sum_{t=1}^{T} s_{b,t}, \qquad s_{b,t} = \frac{1}{D_b}\left\|z_b(\tau_t(x))\right\|_2^2. \tag{6}$$

Grouping retains the branch associated with each latent coordinate, so the final score can combine selected observations:

$$S_{\text{final}}(x) = \frac{1}{2}S_{\text{RAW}}(x) + \frac{1}{2}S_{\text{LL}}(x). \tag{7}$$

Higher scores indicate greater abnormality. The log-determinant remains in the training objective but is excluded from this ranking statistic. HF contributes to shared-parameter learning without entering the final score directly. The resulting separation lets the model learn from detailed normal structure while choosing which branch energies determine test-image ordering. Section 4.4 examines this choice with parameters held fixed.

AUROC is computed from the final image-level scores after view averaging. Appendix A.6 defines energy and energy-plus-Jacobian scores for the same branch set, separating the effect of score formation from the effect of branch selection.

4 Experiments

4.1 Datasets and Evaluation

InsPLAD-fault evaluates real UAV power-line inspection (Vieira e Silva et al., 2023). Following the five-category setting of AttentDifferNet (Vieira e Silva et al., 2024), we use Glass Insulator, Lightning Rod Suspension, Polymer Insulator Upper Shackle, Vari-grip, and Yoke Suspension. Training contains 9,006 normal instances; testing contains 2,254 normal and 402 anomalous instances. Faults include missing insulator units, corrosion, and bird nests. Figure 2 illustrates this variation; Appendix D gives category counts and fault types. MVTec AD supplies ten object and five texture categories from controlled industrial settings (Bergmann et al., 2019).

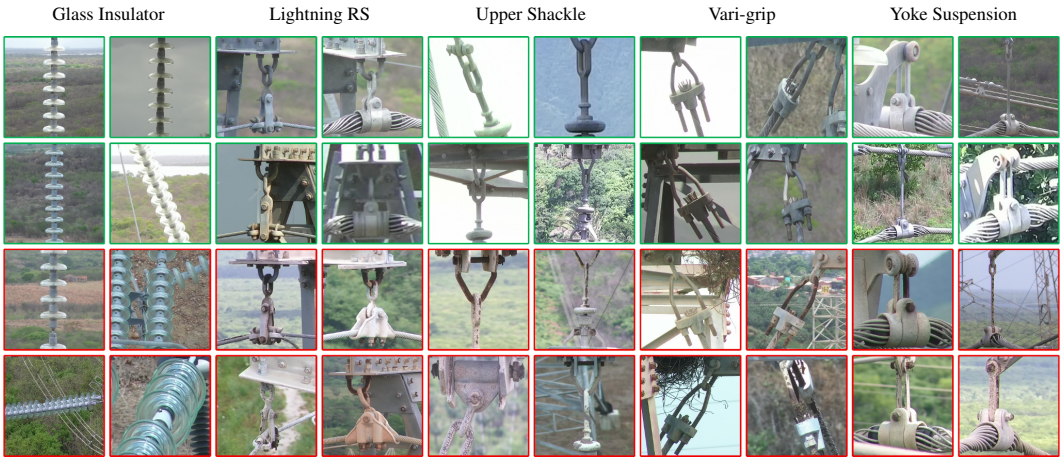

Figure 2: Examples from the five InsPLAD-fault categories. Rows 1 and 2 show normal samples with green borders. Rows 3 and 4 show faulty samples with red borders.

We compute image-level AUROC separately by category, average WBGFlow's two training runs, and macro-average equally over categories. We do not pool uncalibrated category scores or average model predictions before computing AUROC.

4.2 Implementation Details

WBGFlow uses a shared frozen ImageNet-pretrained AlexNet, eight grouped coupling blocks, and 32-dimensional branch embeddings in each block. The main InsPLAD result averages two independent 100-epoch training runs; batch size and learning rate follow DifferNet (Rudolph et al., 2021). Locally evaluated InsPLAD methods use the same training and evaluation protocol. We did not obtain a runnable official training and evaluation implementation of AttentDifferNet (Vieira e Silva et al., 2024), so its local reproduction result is not included.

Images are bilinearly resized to 448×448, rotated by angles uniform in $[-180°, 180°]$, and ImageNet-normalized before branch construction. Rotation uses nearest-neighbor interpolation, a fixed canvas, and black padding; no test-time color jitter, random cropping, or flipping is used. The default view count is $T = 64$.

Component and embedding ablations examine architectural changes through retraining. View-count and scoring analyses instead keep model parameters fixed and use matched rotation sequences, so their comparisons isolate inference choices within each experimental series. Appendix B records the training runs, augmentation settings, and timing configuration. Checkpoint selection used test AUROC; fixing weights for the inference analyses does not remove this earlier selection dependency.

4.3 Quantitative Results

4.3.1 InsPLAD-fault

Table 1 compares the methods under the same training and evaluation protocol. WBGFlow achieves a mean AUROC of 94.62% over two training runs, the highest result in the table. DifferNet obtains 92.46%. The improvement is 2.16 percentage points, with gains in all five categories. WBGFlow also ranks first on Lightning Rod Suspension, Upper Shackle, Vari-grip, and Yoke Suspension.

Table 1: Image-level AUROC (%) on InsPLAD-fault under the same training and evaluation protocol. WBGFlow entries average two independent 100-epoch runs with different random seeds. Avg. is the unweighted five-category macro-average. Values are shown to two decimal places; bold denotes the best result in each column.

Method	Glass Ins.	Light RS	Upper Sha.	Vari Grip	Yoke Sus.	Avg.
FastFlow	70.16	82.02	77.43	65.54	71.48	73.33
RD++	86.21	97.54	83.67	93.85	92.46	90.75
CS-Flow	85.73	96.60	88.40	91.53	90.70	90.59
CFLOW-AD	82.22	95.52	86.60	90.37	83.87	87.72
PatchCore	78.44	85.11	81.02	91.92	58.06	78.91
Dinomaly	**87.73**	92.15	89.19	92.86	93.42	91.07
INP-Former	86.02	89.74	82.95	92.78	88.26	87.95
ReContrast	87.10	96.50	83.30	92.10	91.10	90.02
MSFlow	85.74	92.79	83.81	74.62	87.02	84.80
DifferNet	82.81	99.08	92.42	91.20	96.77	92.46
WBGFlow (ours)	85.23	**99.54**	**95.48**	**94.74**	**98.09**	**94.62**

Comparison across technical routes. The comparison spans several ways of forming anomaly evidence. WBGFlow improves the macro-average over PatchCore's local memory retrieval (Roth et al., 2022) by 15.71 percentage points and over RD++'s reverse distillation (Tien et al., 2023) by 3.87 points. Its macro-average also exceeds those of the evaluated reconstruction methods, including Dinomaly (Guo et al., 2025), INP-Former (Luo et al., 2025), and ReContrast (Guo et al., 2023). The common result is improved image-level ranking across heterogeneous inspection assets. WBGFlow approaches this task through explicit observation groups and their conditional densities, making branch identity and score contribution part of the model design.

DifferNet and WBGFlow both use a frozen encoder and a normalizing flow (Rudolph et al., 2021). WBGFlow changes how the input observations are organized and processed. It uses frequency branches, keeps their features in separate groups, and supplies branch identities to shared coupling subnetworks. Compared with DifferNet, AUROC increases by 3.06 percentage points on Upper Shackle and 3.54 points on Vari-grip. WBGFlow also has a higher macro-average than FastFlow, CFLOW-AD, CS-Flow, and MSFlow. Section 4.4 examines the component changes and scoring choices.

Glass Insulator. Missing discs change both local shape and the spacing between repeated units. Dinomaly uses spatial feature reconstruction (Guo et al., 2025) and achieves 87.73% on Glass Insulator. WBGFlow obtains 85.23%. WBGFlow pools spatial features into global descriptors before density modeling. This pooling may lose information about disc layout and make changes in spacing harder to represent.

4.3.2 MVTec AD

Table 2 groups the MVTec AD results by technical approach. WBGFlow achieves 95.78% AUROC. This is 0.88 percentage points above the 94.90% reported for DifferNet (Rudolph et al., 2021). The WBGFlow result comes from the earlier experiment series. The fixed-checkpoint analyses reported in this paper were conducted on InsPLAD-fault.

Table 2: MVTec AD comparison by technical approach (mean image-level AUROC over 15 categories, %). Technical route identifies the principal modeling or scoring mechanism. C/M denotes category-specific/unified multi-class training. Literature baselines use their reported results; WBGFlow is a previously recorded two-run mean, separate from the current InsPLAD checkpoint analysis.

Method	Technical route	Setting	I-AUROC
PatchCore (Roth et al., 2022)	Memory bank (patch retrieval)	C	99.10
RD++ (Tien et al., 2023)	Distillation (reverse)	C	99.40
Dinomaly (Guo et al., 2025)	Reconstruction (Transformer)	M	99.60
INP-Former (Luo et al., 2025)	Reconstruction (intrinsic prototypes)	M	99.67
ReContrast (Guo et al., 2023)	Reconstruction (contrastive)	M	98.20
CS-Flow (Rudolph et al., 2022)	Flow (cross-scale)	C	98.70
CFLOW-AD (Gudovskiy et al., 2022)	Flow (conditional)	C	98.26
FastFlow (Yu et al., 2021)	Flow (2D feature maps)	C	99.40
MSFlow (Zhou et al., 2025)	Flow (multiscale fusion)	C	99.70
DifferNet (Rudolph et al., 2021)	Flow (global descriptors)	C	94.90
WBGFlow (ours)	Flow (branch-conditioned groups)	C	95.78

4.3.3 Qualitative Results

Figure 3 shows an input image and its saliency map for each inspection category. We use input gradients to visualize score sensitivity (Simonyan et al., 2014). Responses appear near the missing disc in the Glass Insulator example and around connectors, shackle arms, and clamps in the other examples. The gradient target is the final RAW+LL score in Equation (7). HF affects the learned shared weights through Equation (5), but its score is excluded from the gradient target. These maps show sensitivity to input pixels, not defect segmentation. Appendix E describes the procedure and compares the methods under the same visualization settings.

4.4 Ablation Study

Tables 3 and 4 report ablations on Upper Shackle, selected based on its results in the earlier experiments. We retrain the models to test grouping and branch embeddings. We then keep model weights fixed to examine scoring rules and the number of test views.

Preserving observation identity. With RAW/LL/HF inputs, grouped coupling raises AUROC from 93.81% in (B) to 94.41% in (C). Adding branch embeddings raises it to 95.22% in (D). Grouping

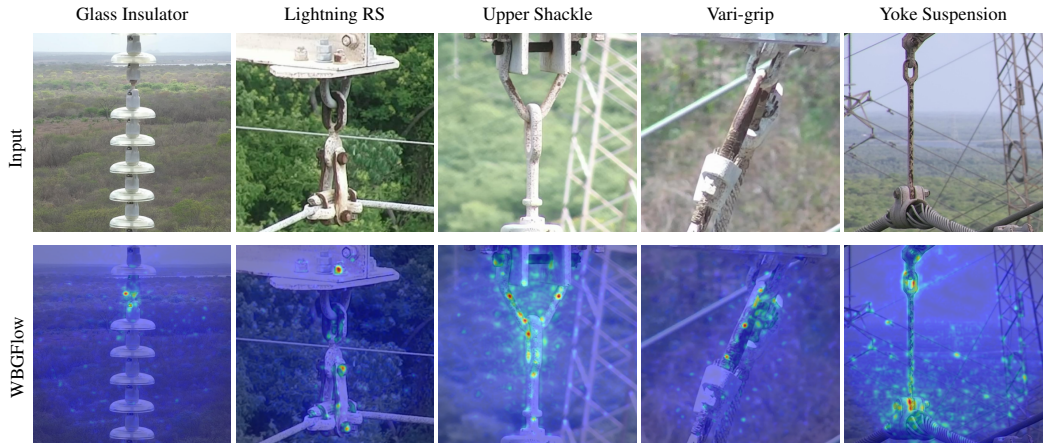

Figure 3: Input images (top) and WBGFlow saliency maps (bottom) for the five InsPLAD-fault categories. Gradients are computed from the RAW+LL score in Equation (7). Appendix E provides the procedure and further comparisons.

Table 3: Component ablations on Upper Shackle (image-level AUROC, %).

ID	LL	HF	Grouped Flow	Branch Emb.	RAW+LL Score	AUROC
A	✗	✗	✗	✗	✗	92.42
B	✓	✓	✗	✗	✗	93.81
C	✓	✓	✓	✗	✗	94.41
D	✓	✓	✓	✓	✗	95.22
E	✓	✓	✓	✓	✓	**96.14**
F	✓	✗	✓	✓	✓	94.30

keeps each branch's latent coordinates separate. The embedding tells the shared subnetwork which branch it is processing. The subnetwork can then predict branch-dependent scales and shifts while using the same weights across branches.

Conditioning shared transformations. Increasing the embedding dimension from 0 to 32 raises S1/S2/S3 AUROC from 92.94% to 95.73% and RAW/LL/HF from 94.81% to 96.14% (Table 4). The benefit under both observation organizations is consistent with identity conditioning adapting shared transformations to different sources.

Table 4: Embedding ablations on Upper Shackle (AUROC, %). (a) Original grouped S1/S2/S3; (b) RAW/LL/HF with RAW+LL scoring.

(a) Original S1/S2/S3

Embedding dimension	AUROC
0	92.94
8	94.38
16	94.64
32	**95.73**

(b) RAW/LL/HF

Embedding dimension	AUROC
0	94.81
8	95.27
16	95.52
32	**96.14**

The flow state remains 768-dimensional because identity vectors enter only the parameter predictors. A dimension of 32 is best among the tested settings. Scoring is fixed within each panel; the zero-dimensional RAW/LL/HF row uses RAW+LL scoring, whereas C averages all three branches.

Separating training observations from ranking evidence. Configuration E uses RAW+LL scoring and reaches 96.14%, compared with 95.22% for D, which averages all three branch scores. Configuration F keeps RAW+LL scoring but removes HF from training. Its AUROC falls to 94.30%, a decrease of 1.84 percentage points from E. HF is excluded from the final score in both E and F. In E, its training loss still updates the shared weights used by RAW and LL, as shown in Equation (5). Appendix C.1 lists the configurations.

Checking the scoring mechanism at fixed parameters. With model weights fixed, RAW+LL scoring achieves 94.320%, compared with 93.381% for the equal average of RAW, LL, and HF. Excluding the HF score improves all five categories. RAW-only scoring reaches 94.363%. Adding LL improves Glass Insulator, Lightning Rod Suspension, and Yoke Suspension, but lowers AUROC on Upper Shackle and Vari-grip.

Energy scoring also gives higher macro-average AUROC than the tested NLL-form scores. The results are 92.474% versus 86.486% for DifferNet and 94.320% versus 89.478% for WBGFlow. The log-determinant remains part of training in these comparisons. Appendix B.3 reports the category results and the fixed scoring weights.

Table 5: Registered parameter counts in millions (M). Shared parameters are counted once. Both methods exclude the unused AlexNet classifier and include registered legacy BatchNorm affine parameters. Reductions are calculated from unrounded counts.

Method	Frozen encoder	Flow	Total
DifferNet	2.470	172.089	174.559
WBGFlow (ours)	2.470	**147.958**	**150.427**
Reduction	0.00%	14.02%	13.82%

Parameter count and computation. Table 5 compares the registered parameters of DifferNet and WBGFlow. Both methods use the same frozen encoder. The reduction therefore comes from the flow module, whose parameter count decreases from 172.089 million to 147.958 million. The total parameter count decreases by 13.82%.

The grouped coupling subnetworks process features within each branch and share their weights across RAW, LL, and HF. Branch embeddings allow these shared subnetworks to predict different scales and shifts for each input source. This avoids assigning a separate set of coupling weights to every branch. Together with the InsPLAD-fault results, the comparison shows that the AUROC improvement is obtained with fewer registered parameters. Appendix B.4 gives the detailed parameter accounting.

Test-time view aggregation provides another way to adjust computational cost without retraining. With fixed checkpoints, reducing the view count from 64 to 16 changes AUROC from 94.320% to 94.308%. Model-stage throughput increases from 8.698 to 34.455 images/s. Appendix B.2 reports the complete view-budget comparison, and Appendix B.4 specifies the timing conditions.

5 Conclusion

We introduced WBGFlow for anomaly detection in uncontrolled power-line inspection. The model keeps RAW, LL, and HF features in separate groups and uses branch embeddings to condition shared coupling subnetworks. All three branches contribute to training, while only RAW and LL contribute to the final score. WBGFlow achieves a mean AUROC of 94.62% over two training runs on InsPLAD-fault. It exceeds DifferNet by 2.16 percentage points and improves all five categories. It also reaches 95.78% on MVTec AD. The ablations show how grouping, branch conditioning, and score selection contribute to the inspection results.

AI Use Statement

Generative AI tools were used to assist with translation, language editing, LaTeX typesetting, and figure presentation.

REFERENCES

Akshatha Arodi, Margaux Luck, Jean-Luc Bedwani, Aldo Zaimi, Ge Li, Nicolas Pouliot, Julien Beaudry, and Gaétan M. Caron. CableInspect-AD: An expert-annotated anomaly detection dataset. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, pp. 64703–64716, 2024.

Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 128–138, 2024.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9584–9592, 2019.

Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022.

Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 475–489. Springer, 2021.

Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9737–9746, 2022.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Cory Gardner, Aendri Dhiren, Byungseok Min, and Tae-Hyuk Ahn. Wavelet-enhanced PaDiM for industrial anomaly detection. *IEEE Access*, 14:37702–37718, 2026.

Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 98–107, 2022.

Jia Guo, Shuai Lu, Lize Jia, Weihang Zhang, and Huiqi Li. ReContrast: Domain-specific anomaly detection via contrastive reconstruction. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10721–10740, 2023.

Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20405–20415, 2025.

Lars Heckler-Kram, Jan-Hendrik Neudeck, Ulla Scheler, Rebecca König, and Carsten Steger. The MVTec AD 2 dataset: Advanced scenarios for unsupervised anomaly detection. *International Journal of Computer Vision*, 134(4), 2026. Article 175.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19606–19616, 2023.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20578–20589, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105, 2012.

Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9664–9674, 2021.

Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. MuSc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *International Conference on Learning Representations*, 2024.

Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20402–20411, 2023.

Wei Luo, Yunkang Cao, Haiming Yao, Xiaotian Zhang, Jianan Lou, Yuqi Cheng, Weiming Shen, and Wenyong Yu. Exploring intrinsic normal prototypes within a single image for universal anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9974–9983, 2025.

Stéphane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. FcaNet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 783–792, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but DifferNet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1907–1916, 2021.

Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1088–1097, 2022.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402, 2018.

Dahu Shi, Chengshen He, Shaochen Zhang, Bo Qian, Xiaochen Quan, Wencong Zhang, and Xing Wei. Omni-AD: A large-scale and versatile benchmark for industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14157–14166, 2026.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations, Workshop Track*, 2014.

Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan T. M. Duong, Chanh D. Tr. Nguyen, and Steven Q. H. Truong. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24511–24520, 2023.

André Luiz Vieira e Silva, Francisco Simões, Danny Kowerko, Tobias Schlosser, Felipe Battisti, and Veronica Teichrieb. Attention modules improve image-level anomaly detection for industrial inspection: A DifferNet case study. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 8246–8255, 2024.

André Luiz Buarque Vieira e Silva, Heitor de Castro Felix, F. P. M. Simões, Veronica Teichrieb, Michel dos Santos, Hemir Santiago, Virginia Sgotti, and Henrique Lott Neto. InsPLAD: A dataset and benchmark for power line asset inspection in UAV images. *International Journal of Remote Sensing*, 44(23):7294–7320, 2023.

Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-IAD: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22883–22892, 2024.

Xu Wang, Jingyuan Zhuo, Zhiyuan You, Zhiyu Tan, Yikuan Yu, Siyu Wang, and Xinyi Le. ReinAD: Towards real-world industrial anomaly detection with a comprehensive contrastive dataset. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38, pp. 21958–21976, 2025.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *European Conference on Computer Vision*, pp. 3–19. Springer, 2018.

Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4571–4584, 2022.

Jason J. Yu, Konstantinos G. Derpanis, and Marcus A. Brubaker. Wavelet flow: Fast training of high resolution normalizing flows. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6184–6196, 2020.

Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. FastFlow: Unsupervised anomaly detection and localization via 2D normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.

Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRAEM—a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8330–8339, 2021.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *International Conference on Learning Representations*, 2024.

Yixuan Zhou, Xing Xu, Jingkuan Song, Fumin Shen, and Heng Tao Shen. MSFlow: Multiscale flow-based framework for unsupervised anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2437–2450, 2025.

A Mathematical and Implementation Details

A.1 Formal Problem Definition

For a given asset category, the normal training set is

$$\mathcal{D}_{\mathrm{tr}} = \{x_i\}_{i=1}^{N}, \qquad y_i = 0 \quad (i = 1, \ldots, N). \tag{A.1}$$

where $x_i \in \mathbb{R}^{3\times H\times W}$ is an RGB image and $y_i = 0$ indicates that no anomalous samples are used for training. The test set

$$\mathcal{D}_{\mathrm{te}} = \{(x_j, y_j)\}_{j=1}^{M}, \qquad y_j \in \{0, 1\}. \tag{A.2}$$

contains both normal and anomalous images. The model learns an anomaly scoring function $S(x) \in \mathbb{R}$, with a higher score indicating a greater departure from the normal distribution. We do not learn a fixed decision threshold; instead, image-level AUROC measures the quality of normal/anomalous rankings across all possible thresholds.

A.2 Haar Indexing and Directional Components

For an input with even height and width, the four subsampled tensors use zero-based spatial indices:

$$(x_{pq})_{c,i,j} = x_{c,\,2i+p,\,2j+q}, \qquad p, q \in \{0, 1\}. \tag{A.3}$$

Here, $c \in \{1, 2, 3\}$, $i = 0, \ldots, H/2 - 1$, and $j = 0, \ldots, W/2 - 1$. The three directional components are

$$\begin{aligned} x_{\mathrm{LH}} &= \frac{-x_{00} - x_{01} + x_{10} + x_{11}}{2}, \\ x_{\mathrm{HL}} &= \frac{-x_{00} + x_{01} - x_{10} + x_{11}}{2}, \\ x_{\mathrm{HH}} &= \frac{x_{00} - x_{01} - x_{10} + x_{11}}{2}. \end{aligned} \tag{A.4}$$

Together with the low-frequency average in Equation (1) and high-frequency energy in Equation (2), these define the observation branches:

$$\mathcal{B}(x) = \{x_{\mathrm{RAW}}, x_{\mathrm{LL}}, x_{\mathrm{HF}}\}, \qquad x_{\mathrm{RAW}} = x. \tag{A.5}$$

The operations apply independently to the three color channels. RAW has resolution 448×448, while LL and HF have resolution 224×224.

A.3 Feature and State Definitions

Let E be the shared frozen AlexNet feature extractor. For each branch, the feature map and globally pooled descriptor are

$$F_b = E(x_b), \qquad v_b = \mathrm{GAP}(F_b) \in \mathbb{R}^{256}. \tag{A.6}$$

Concatenation along the feature dimension gives

$$v = [v_{\mathrm{RAW}}; v_{\mathrm{LL}}; v_{\mathrm{HF}}] \in \mathbb{R}^{768}. \tag{A.7}$$

A.4 Splitting, Identity Dimensions, and Scaling

The group-preserving permutation maintains the three observation blocks:

$$\widetilde{v} = \Pi(v) = \big[\pi_{\mathrm{RAW}}(v_{\mathrm{RAW}}); \pi_{\mathrm{LL}}(v_{\mathrm{LL}}); \pi_{\mathrm{HF}}(v_{\mathrm{HF}})\big]. \tag{A.8}$$

After the within-group permutation in Equation (A.8), each branch is split into equal halves:

$$\widetilde{v}_b = [u_{b,1}; u_{b,2}], \qquad u_{b,1}, u_{b,2} \in \mathbb{R}^{128}. \tag{A.9}$$

Each block has a separate identity vector for each branch. Omitting the block index,

$$e_b \in \mathbb{R}^{d_e}, \qquad d_e = 32. \tag{A.10}$$

Identity concatenation occurs after permutation and splitting, giving a 160-dimensional input to each shared coupling subnetwork. Embeddings condition the scale and shift predictions but do not enter the invertible state. The state therefore remains 768-dimensional. Each subnetwork outputs 256 values; the composed operation $\mathrm{split}(g_{\theta,j}(\cdot))$ returns 128 log-scale and 128 shift parameters. The scaling operator is

$$a(s) = \exp\big(C_\alpha(s)\big), \qquad \alpha = 3. \tag{A.11}$$

Here, C_α is the arctangent soft-clamping function used by the implementation. The bounded log-scale limits exponential scaling. The two affine steps are given in Equations (3) and (4).

In Figure 1, $E^{(k)}$ contains the three identity vectors of block k. Each conditioning node C concatenates a branch's half-vector with its corresponding embedding, yielding a subnetwork input dimension of $128 + 32 = 160$. Within-group permutation and splitting occur before identity concatenation. The embedding does not enter the invertible state, whose total dimension remains 768.

Multiplication and addition nodes apply scaling and shifting, respectively, and the final C node concatenates the two output halves within each branch. In the figure, X_1, X_2 denote input half-vectors, Y_1, Y_2 denote their updated counterparts, and S_j, T_j denote affine parameters. These symbols are distinct from anomaly scores and from the original S1/S2/S3 scale indices.

A.5 Gaussian Prior and Change-of-Variables Derivation

Let $\theta = \{\theta_k\}_{k=1}^{8}$ denote the weights shared across branches within each block, and let $\mathcal{E} = \{e_b^{(k)}\}_{b,k}$ denote the per-block branch embeddings. The complete invertible mapping determined by these parameters is $f_{\theta,\mathcal{E}} : \mathbb{R}^D \to \mathbb{R}^D$, where $D = 768$. Its latent output is

$$z(x) = f_{\theta,\mathcal{E}}(v(x)) \in \mathbb{R}^D. \tag{A.12}$$

We use a standard Gaussian prior with density

$$p_Z(z) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\|z\|_2^2\right). \tag{A.13}$$

Let $J_f(v) = \partial f_{\theta,\mathcal{E}}(v)/\partial v$ be the Jacobian of the flow. The change-of-variables formula (Dinh et al., 2017) gives the input feature density:

$$\log p_V(v) = \log p_Z(f_{\theta,\mathcal{E}}(v)) + \log\left|\det J_f(v)\right|. \tag{A.14}$$

Ignoring constants independent of the model parameters, the implemented negative log-likelihood normalized by dimension is

$$\mathcal{L}_{\mathrm{NF}} = \frac{1}{D}\,\mathbb{E}_{x\sim\mathcal{D}_{\mathrm{tr}}}\left[\frac{1}{2}\|z(x)\|_2^2 - \log\left|\det J_f(v(x))\right|\right]. \tag{A.15}$$

Both $v(x)$ and $z(x)$ are computed from a normal image x. The energy and log-determinant terms jointly fit the standard Gaussian prior, with the latter accounting for volume changes under the invertible transformation.

Let $J_b(x)$ denote the Jacobian of the branch mapping evaluated at $v_b(x)$. The branch negative log-likelihood is

$$\mathcal{L}_b = \frac{1}{D_b}\,\mathbb{E}_{x\sim\mathcal{D}_{\mathrm{tr}}}\left[\frac{1}{2}\|z_b(x)\|_2^2 - \log\left|\det J_b(x)\right|\right]. \tag{A.16}$$

The branch objective is defined in Equation (A.16). Equation (5) gives the loss decomposition and shared-parameter gradient. These relations show how HF contributes to learning weights also used by RAW and LL, while branch forward paths remain separate for fixed parameters.

A.6 Energy and NLL Scoring Definitions

For branch b, latent dimension $d = 256$, and view t, define the half-energy and negative normalized log-determinant by

$$E_{t,b} = \frac{\|z_{t,b}\|_2^2}{2d}, \qquad V_{t,b} = -\frac{1}{d}\log\left|\det J_{f_b}(v_{t,b})\right|. \tag{A.17}$$

Under an invertible feature mapping and a standard Gaussian prior,

$$-\frac{1}{d}\log p_b(v_{t,b}) = E_{t,b} + V_{t,b} + \frac{1}{2}\log(2\pi). \tag{A.18}$$

For a chosen branch set $\mathcal{S}$, the fixed-weight score is

$$\begin{aligned} S^{(T)}_{\mathcal{S},\lambda}(x) &= \frac{1}{T|\mathcal{S}|}\sum_{t=1}^{T}\sum_{b\in\mathcal{S}}(E_{t,b} + \lambda V_{t,b}), \\ S^{(T)}_{\mathcal{S},\mathrm{E}} &= 2S^{(T)}_{\mathcal{S},0}, \qquad S^{(T)}_{\mathcal{S},\mathrm{NLL}} = S^{(T)}_{\mathcal{S},1}. \end{aligned} \tag{A.19}$$

The NLL score omits the sample-independent Gaussian constant. The energy score with $\mathcal{S} = \{\mathrm{RAW}, \mathrm{LL}\}$ equals Equation (7). At $\lambda = 0$, the positive factor of two changes score magnitudes but not AUROC. A joint WBGFlow score uses all three branches. Group-preserving transformations satisfy

$$\log|\det J_f| = \sum_{b\in\{\mathrm{RAW},\mathrm{LL},\mathrm{HF}\}} \log|\det J_{f_b}|. \tag{A.20}$$

Selected RAW+LL NLL uses the corresponding two branch log-determinants, not a redistributed joint log-determinant. DifferNet uses its complete 768-dimensional representation and joint log-determinant, normalized by 768; globally mixed latent slices are not interpreted as scale-specific scores. The weights $\lambda \in \{0, 0.25, 0.5, 1, 2\}$ are fixed, not learned, and are not selected separately for each test category.

Numerical interpretation. The density formulas require invertible transformations. Clipping the transformed state can violate this requirement. The implementation includes magnitude clipping, but the available evaluation records do not show whether it was triggered. We therefore call the recorded energy-plus-Jacobian statistic an *NLL-form score*. Its interpretation as a negative log-likelihood requires clipping to remain inactive. The arctangent clamp has a different role. It limits predicted log-scales while keeping the affine scale factors positive.

B Additional Scoring and Inference Analysis

B.1 Protocol, Data Scope, and Run Statistics

WBGFlow's two main training runs produce macro-average AUROCs of 94.914% and 94.320%, respectively. Their mean is 94.617%, displayed as 94.62% in Table 1; each category entry is averaged in the same way. The comparison uses the common protocol stated in the main text, including DifferNet's 92.46% baseline.

The fixed-checkpoint inference analysis evaluates all 2,656 test images using the seed1 checkpoint of each method and category, corresponding to the second main WBGFlow run, with test-augmentation seed 0. Component and embedding ablations use the earlier training runs with different seeds; effects are compared within each series. The fixed-checkpoint results are 94.320% for WBGFlow and 92.474% for DifferNet, a 1.846-point difference (Table B.1). The six view budgets use nested prefixes of the same rotation sequence, matched between methods; $T = 1$ is a random rotation, not an identity transform.

Table B.1: Matched-checkpoint InsPLAD-fault results: image-level AUROC (%), $T = 64$. Checkpoints, test images, and rotations follow the protocol in Appendix B. Differences use unrounded values; Glass is effectively tied.

Asset category	DifferNet	WBGFlow	Δ (pp)
Glass Insulator	84.351	84.360	+0.010
Lightning Rod Suspension	99.436	99.538	+0.103
Polymer Insulator Upper Shackle	92.653	94.823	+2.169
Vari-grip	91.726	94.761	+3.035
Yoke Suspension	94.202	98.116	+3.914
Macro-average	92.474	**94.320**	**+1.846**

Input construction resizes RGB images to 448×448 by bilinear interpolation and applies a centered rotation drawn uniformly from $[-180°, 180°]$ with nearest-neighbor interpolation, no canvas expansion, and zero padding. Floating-point values in $[0, 1]$ are normalized using mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. Branches are constructed after normalization. There is no additional color jitter, random cropping, flipping, or translation. At $T = 64$, the three branches produce 192 feature-extractor inputs per image. The number of forward calls depends on how these inputs are batched.

The main WBGFlow result uses two training runs. Let $A(c, r)$ be the AUROC for category c in run r, where $r \in \{1, 2\}$. The reported means are

$$\overline{A}_c = \frac{1}{2} \sum_{r=1}^{2} A(c, r), \qquad \overline{A} = \frac{1}{C} \sum_{c=1}^{C} \overline{A}_c. \tag{B.1}$$

We use $C = 5$ for InsPLAD-fault and $C = 15$ for MVTec AD. AUROC is calculated separately for each run and then averaged. Predictions from the two models are not combined. The inference tables use the fixed checkpoints described above.

We selected the checkpoints using test AUROC. The same checkpoints are used across the inference settings. This keeps those comparisons consistent, but the results still depend on test-based checkpoint selection.

B.2 Test-View Budget

This experiment varies the amount of test-time aggregation while keeping the learned model fixed. WBGFlow has higher macro-average AUROC than DifferNet at all six tested budgets. At $T = 16$, its 94.308% is 0.012 points below the $T = 64$ result, while model-stage throughput is 34.455 rather than 8.698 images/s, a factor of 3.96. The $T = 32$ result is 94.355%; the ordering is not monotonic, indicating that additional views do not consistently improve ranking under this protocol.

Against DifferNet at $T = 64$, WBGFlow at $T = 16$ gains 1.834 points at 3.02 times the model-stage throughput. This comparison explicitly uses different view budgets. At a matched $T = 64$, WBGFlow

Table B.2: View-budget sensitivity using the fixed checkpoints in Table B.1. AUROC is the five-category macro-average; throughput counts original images including all T views. $T = 1$ is a random rotation.

	AUROC (%)		Model images/s	
T	DifferNet	WBGFlow	DifferNet	WBGFlow
1	83.354	91.488	176.056	135.145
4	88.919	93.415	79.469	60.399
8	90.967	93.834	55.460	45.776
16	92.011	94.308	43.746	34.455
32	92.690	94.355	22.783	17.417
64	92.474	94.320	11.397	8.698

is slower, as detailed in Appendix B.4. Rotation interpolation, crop boundaries, and black padding also change the observations, so this experiment measures aggregation under the stated protocol rather than isolating rotation invariance.

B.3 Scoring Weights and Branch Scope

We use the fixed-checkpoint results to test the consequence of preserving separate scoring paths. In this section, NLL denotes the NLL-form energy-plus-Jacobian score defined in Appendix A.6; its exact-density interpretation has the conditions stated there.

Table B.3: Fixed-checkpoint scoring analysis at $T = 64$: per-category AUROC (%). Selected WBGFlow uses RAW+LL; joint WBGFlow uses RAW+LL+HF. Energy has the same ranking as $\lambda = 0$, and NLL uses $\lambda = 1$. Parameters and test views remain fixed.

Method / scope	Score	Glass	Light RS	Upper Sha.	Vari-grip	Yoke	Macro
DifferNet / full	Energy	84.351	99.436	92.653	91.726	94.202	92.474
	$\lambda = 0.25$	78.799	99.214	90.146	91.860	94.950	90.994
	$\lambda = 0.5$	75.948	98.872	87.426	90.398	92.645	89.058
	NLL	73.779	97.778	83.892	86.645	90.339	86.486
	$\lambda = 2$	72.322	95.778	80.417	81.373	88.661	83.710
WBGFlow / RAW+LL	Energy	84.360	99.538	94.823	94.761	98.116	94.320
	$\lambda = 0.25$	83.178	99.641	92.649	94.286	99.045	93.760
	$\lambda = 0.5$	79.801	99.385	90.033	92.224	98.811	92.051
	NLL	76.213	98.598	86.341	88.415	97.822	89.478
	$\lambda = 2$	73.433	97.094	82.436	84.076	96.263	86.661
WBGFlow / all three	Energy	83.098	98.479	94.197	93.528	97.604	93.381
	$\lambda = 0.25$	80.562	97.983	90.818	86.834	97.430	90.725
	$\lambda = 0.5$	77.053	96.718	87.580	80.860	95.785	87.599
	NLL	73.785	94.308	84.009	73.961	93.531	83.919
	$\lambda = 2$	71.222	90.171	80.930	68.777	91.411	80.502

Energy outperforms NLL in all five categories for both models. DifferNet's macro-average falls by 5.987 points when moving from energy to NLL; the selected RAW+LL WBGFlow score falls by 4.842 points. The negative log-determinant can change normal/anomalous ordering even though it is required by the training density objective. At $\lambda = 0.25$, WBGFlow improves on Lightning RS and Yoke but has a lower macro-average than energy. These are fixed-weight inference comparisons, not learned fusion or evidence for removing the log-determinant during training.

For the jointly trained three-branch model, excluding the direct HF energy improves all five categories and raises the macro-average from 93.381% to 94.320%. RAW-only scoring is 94.363%, exceeding RAW+LL by 0.043 points. Fusion helps Glass, Lightning RS, and Yoke but lowers Upper Shackle and Vari-grip. Individual branch scores still come from the three-branch-trained model, and do not isolate the contribution of a branch's training loss.

B.4 Parameter Accounting and Timing

For a subnetwork with input dimension I, output dimension O, and three hidden layers of width H, the registered count is

$$P(I, O; H) = H(I + 1) + 2H(H + 1) + O(H + 1) + 2I. \tag{B.2}$$

The final term counts registered legacy BatchNorm affine parameters that are not used in the current forward path. With $H = 2048$, eight blocks and two subnetworks per block give

$$\begin{aligned} P_{\mathrm{NF,D}} &= 16P(384, 768; 2048) = 172{,}089{,}344, \\ P_{\mathrm{NF,W}} &= 16P(160, 256; 2048) + 8 \cdot 3 \cdot 32 \\ &= 147{,}957{,}504. \end{aligned} \tag{B.3}$$

Table B.4: Registered parameters, counting shared parameters once. Both methods exclude the unused AlexNet classifier and include registered legacy BatchNorm affine parameters.

Method	Frozen encoder	Flow	Total
DifferNet	2,469,696	172,089,344	174,559,040
WBGFlow	2,469,696	147,957,504	150,427,200

Total registered parameters decrease by 13.82%, and flow parameters by 14.02%; the denominators differ. Identity tables alone contain 768 parameters, but concatenation also changes the first layer. Registered or nominally trainable counts do not imply that every parameter participates in forward computation or receives a gradient.

Timing uses a Tesla T4 under Linux, PyTorch 2.1.2+cu121, torchvision 0.16.2+cu121, and CUDA 12.1. Evaluation uses FP32, original-image batch size one, view chunks of at most 16, deterministic cuDNN, and inference mode, with autocast and TF32 disabled. Each category receives five model-stage warmups and 20 timed repetitions, pooled over 100 original images. Model timing includes all branch encoders, flow operations, and selected-score aggregation, but excludes loading, CPU transforms, and host-to-device transfer.

For R measurements with durations τ_r in seconds and B original images per measurement, pooled throughput Q and amortized time L are

$$Q = \frac{BR}{\sum_{r=1}^{R} \tau_r}, \qquad L = \frac{10^3}{BR} \sum_{r=1}^{R} \tau_r. \tag{B.4}$$

The units are original images/s and ms/original image, respectively; all T views belong to one image. Relative time overhead is

$$\Delta L = \left(\frac{L_{\mathrm{W}}}{L_{\mathrm{D}}} - 1 \right) 100\%. \tag{B.5}$$

End-to-end timing adds loading, CPU transforms, and host-to-device transfer, but excludes checkpoint loading. It covers one pass over the first 32 ordered images per category (160 total), with zero workers. These are subset measurements without repeated-timing error bars. At $T = 64$ with energy scoring, DifferNet and WBGFlow take 957.596 and 966.659 ms/image, respectively (+0.95%). The opposite local trend for NLL end-to-end time is not evidence of acceleration; the current NLL implementation also repeats some subnetwork computation to obtain log-determinants.

The largest PyTorch peak allocated memory over the five categories at $T = 64$ is 1009.229 MiB for DifferNet and 945.275 MiB for WBGFlow.

Table B.5: Model-stage and subset end-to-end timing. Rates count original images including all T views. Model timing pools 100 images; end-to-end (E2E) timing covers one pass over 160 images, not the full test set.

Method	Score	T	Model img/s	Model ms/img	E2E img/s	E2E ms/img
DifferNet	Energy	1	176.056	5.680	18.729	53.393
DifferNet	Energy	4	79.469	12.583	10.695	93.503
DifferNet	Energy	8	55.460	18.031	7.526	132.879
DifferNet	Energy	16	43.746	22.859	3.845	260.073
DifferNet	Energy	32	22.783	43.892	1.958	510.633
DifferNet	Energy	64	11.397	87.742	1.044	957.596
DifferNet	NLL	64	9.072	110.225	1.018	981.951
WBGFlow	Energy	1	135.145	7.399	16.898	59.178
WBGFlow	Energy	4	60.399	16.556	10.081	99.199
WBGFlow	Energy	8	45.776	21.845	6.831	146.384
WBGFlow	Energy	16	34.455	29.023	4.166	240.036
WBGFlow	Energy	32	17.417	57.415	2.023	494.407
WBGFlow	Energy	64	8.698	114.975	1.034	966.659
WBGFlow	NLL	64	7.053	141.774	1.068	936.348

B.5 Extreme Views and Scope of Interpretation

We use the recorded scores without removing outliers, clipping scores, or reversing scores based on labels. A single view dominates the average for some images. One normal Upper Shackle image has a DifferNet energy score of approximately 533,151.395. One view contributes 79.91% of that average. For one anomalous Upper Shackle image, the largest WBGFlow view contribution is 94.27%. Medians, interquartile ranges, and tail quantiles help describe these uneven distributions. AUROC depends on score ordering. A large score alone does not establish a detection error or background interference.

These comparisons use fixed checkpoints and the stated augmentation settings. The saliency maps in Appendix E separately show which image regions affect the score.

C Additional Ablation Details

C.1 Component Configurations

Table 3 compares six configurations under the settings in Section 4.2. A is DifferNet. B uses RAW/LL/HF inputs with a conventional flow that mixes all feature dimensions. C replaces this flow with grouped coupling. D adds 32-dimensional branch embeddings and averages the three branch scores. E trains with all three branches but scores images using RAW and LL. F uses the same scoring rule as E and removes HF from training. D and E compare scoring choices. E and F compare training with and without HF under RAW+LL scoring.

C.2 Embedding Dimension under Two Observation Organizations

The embedding dimension controls the capacity of the conditioning input without changing the 768-dimensional invertible state. Table 4 examines this mechanism under both the original S1/S2/S3 organization and RAW/LL/HF, testing whether the role of identity conditions is restricted to wavelet observations.

Under the original S1/S2/S3 organization, the zero- and 32-dimensional settings achieve 92.94% and 95.73%, respectively, a difference of 2.79 percentage points. The benefit of branch conditioning also appears with scale-based observations, consistent with its role in adapting to statistics from different sources. The networks can retain shared parameters while producing branch-dependent transformations through their identity inputs, without adding the condition vectors to the feature state whose density is modeled.

With RAW/LL/HF training and RAW+LL scoring fixed, embedding dimensions of 0, 8, 16, and 32 yield AUROCs of 94.81%, 95.27%, 95.52%, and 96.14%, respectively, increasing across the four evaluated settings. Identity vectors provide explicit branch conditions to the shared subnetworks, allowing the same weights to adjust scaling and shifting according to the observation source. These results support the role of identity conditioning in adapting shared transformations. A dimension of 32 is the best of the tested settings.

D DATASET DETAILS

InsPLAD contains 10,607 high-resolution UAV images and 17 asset categories, supporting asset detection, fault classification, and anomaly detection on operational power lines (Vieira e Silva et al., 2023). We follow AttentDifferNet's five-category unsupervised setting (Vieira e Silva et al., 2024). Figure 2 shows normal and anomalous examples; Table D.1 gives the splits and fault types for the 11,662 cropped asset instances used here.

Table D.1: Sample counts and anomaly types in the InsPLAD-fault unsupervised anomaly detection subset. The five categories cover six combinations of assets and fault types. For Vari-grip, 63/48 denotes bird-nest/corrosion anomalies, respectively.

Asset category	Anomaly type	Train normal	Test normal	Test anomalous	Total
Glass Insulator	Missing cap	2,298	581	90	2,969
Lightning Rod Suspension	Corrosion	462	117	50	629
Polymer Insulator Upper Shackle	Corrosion	935	235	102	1,272
Vari-grip	Bird's nest / corrosion	477	114	63/48	702
Yoke Suspension	Corrosion	4,834	1,207	49	6,090
Total	–	**9,006**	**2,254**	**402**	**11,662**

The table covers real inspection faults. Vari-grip's 63 bird-nest cases and 48 corrosion cases are jointly counted as 111 anomalous instances (Vieira e Silva et al., 2024). Category-wise AUROC and the unweighted five-category macro-average give equal weight to categories despite their different sample counts.

MVTec AD (Bergmann et al., 2019) is a widely used public benchmark for unsupervised anomaly detection. It contains normal and anomalous images of objects and textures acquired in controlled industrial settings, covering ten object categories and five texture categories. The dataset provides image-level normal/anomalous labels and pixel-level masks. We use only image-level annotations in this work.

Benchmark context. Beyond standard industrial anomaly detection, benchmarks have introduced logical constraints (Bergmann et al., 2022), multiple viewpoints (Wang et al., 2024), comparisons involving real production settings (Wang et al., 2025), and evaluation across multiple tasks (Shi et al., 2026). These settings emphasize different sources of difficulty. We use InsPLAD-fault to assess real inspection imagery and MVTec AD as a complementary controlled benchmark, without treating their absolute AUROCs as interchangeable measures of scene difficulty.

E Additional Qualitative Results and Interpretability Analysis

We use input-gradient saliency maps to examine how anomaly decisions respond to different image regions (Simonyan et al., 2014), following DifferNet's analysis of backpropagating anomaly scores to the input image (Rudolph et al., 2021). DifferNet and WBGFlow use identical settings for view alignment, gradient aggregation, and post-processing to control for differences introduced by the visualization pipeline.

Let $x_t = \tau_t(x)$ be the t-th rotated view, $x_{t,c}$ its c-th color channel, and T the number of views. The per-view branch score $s_{b,t}$ is computed as in Equation (6). For WBGFlow, the per-view fused score s_t and the final image-level anomaly score satisfy

$$\begin{aligned} s_t &= \frac{s_{\mathrm{RAW},t} + s_{\mathrm{LL},t}}{2}, \\ S_{\mathrm{final}}(x) &= \frac{1}{T}\sum_{t=1}^{T} s_t = \frac{S_{\mathrm{RAW}}(x) + S_{\mathrm{LL}}(x)}{2}. \end{aligned} \tag{E.1}$$

The gradient target for each view is therefore consistent with the test-time RAW+LL fusion in Equation (7). HF does not directly enter either s_t or the final anomaly score.

Using s_t as the gradient target for each view, we first inverse-rotate the gradients into the original image coordinates. Absolute gradients are then aggregated across views and color channels, followed by smoothing, squaring, and normalization to obtain the saliency map:

$$H(x) = \mathcal{N}\left(\left[\mathcal{G}\left(\frac{1}{3T}\sum_{t=1}^{T}\sum_{c=1}^{3}\left|\mathcal{A}_t\left(\frac{\partial s_t}{\partial x_{t,c}}\right)\right|\right)\right]^2\right). \tag{E.2}$$

Here, $\mathcal{A}_t$ is the inverse-rotation alignment operator, $\mathcal{G}$ is a smoothing operator, and $\mathcal{N}$ is a normalization operator; absolute values and squares are applied elementwise. Equation (E.1) defines the image-level ranking score, while Equation (E.2) visualizes the spatial sensitivity of that scoring rule.

Reading the comparisons. Figure 3 shows one input and saliency pair for each of the five InsPLAD-fault categories. The appendix contains 20 additional InsPLAD-fault examples and 74 MVTec AD examples. Figure E.1 includes three Lightning Rod Suspension, five Upper Shackle, and two Vari-grip examples. Figure E.2 includes one Vari-grip and nine Yoke Suspension examples. The MVTec AD panels contain five examples per category, except Toothbrush, which has four.

Each triplet shows the input, DifferNet, and WBGFlow from left to right. We retain all supplied examples. Headings, blank padding, and tile borders are cropped consistently across matched triplets. Image and heatmap colors are unchanged. These maps come from the earlier experiments. Their generating checkpoints have not been matched to those used in the current inference tables.

Colors show relative sensitivity within each image. Equation (E.2) uses absolute gradients, so the maps do not show whether a pixel increases or decreases the score. Detection performance is assessed separately using AUROC.

E.1 Responses in Uncontrolled Inspection Scenes

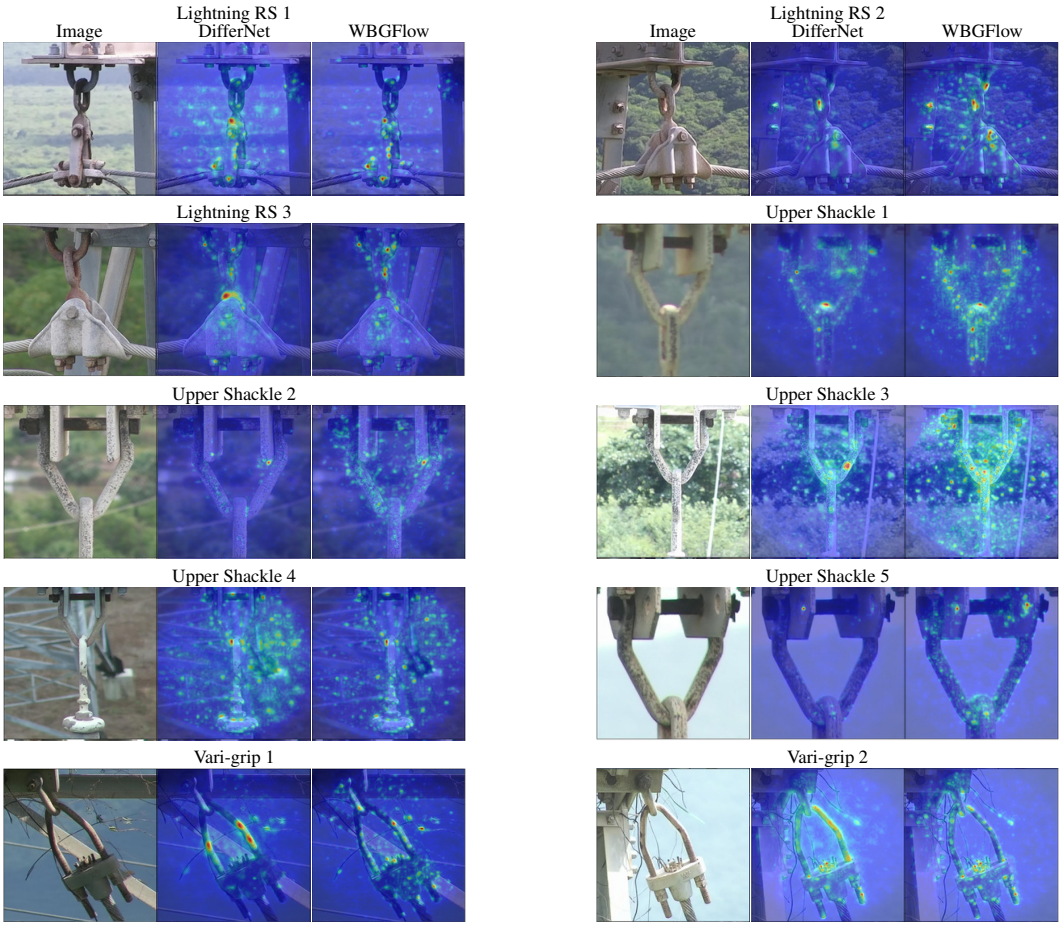

Figure E.1: InsPLAD-fault comparisons: Lightning RS 1–3, Upper Shackle 1–5, and Vari-grip 1–2. Each labelled triplet shows Image, DifferNet, and WBGFlow; sample indices follow the original selection.

Lightning Rod Suspension. Both methods respond to the connecting rings and lower clamp in Figure E.1. In sample 1, WBGFlow responds along the foreground connector and shows fewer scattered responses over vegetation. In sample 3, its strongest responses move toward the corroded upper link. Sample 2 still shows responses around the hardware and background.

Polymer Insulator Upper Shackle. WBGFlow responds along the shackle arms and connecting end. In sample 4, responses to the right of the component are less scattered, while responses on the shackle and lower connection remain. The first three samples show broader responses around the equipment. Sample 3 also retains visible responses on vegetation.

Vari-grip. Both methods respond to the curved rods and fasteners in Figures E.1 and E.2. In sample 2, WBGFlow shows less spread over the sky while retaining responses on the hardware. Sample 3 still responds to the component and the distant scene on its left. Thin branches and nearby structural edges also receive responses in the first two samples.

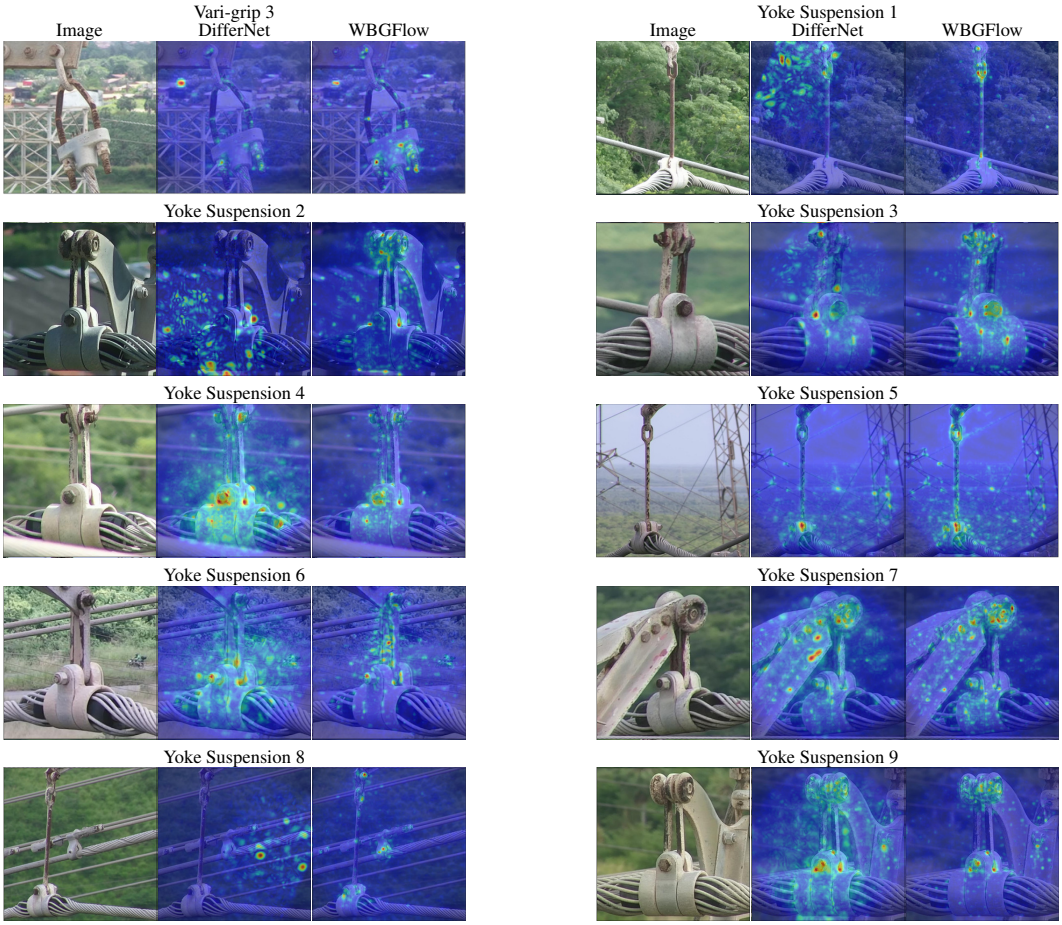

Figure E.2: Additional InsPLAD-fault comparisons: Vari-grip 3 and Yoke Suspension 1–9. The labelled triplets use the same Image, DifferNet, and WBGFlow order.

Yoke Suspension. In sample 1 of Figure E.2, DifferNet responds broadly to foliage. WBGFlow places more response on the upper connection and the thin foreground member. In sample 2, scattered responses below and to the left of the equipment are reduced. Responses around the lower clamp are also less spread out in samples 4 and 9. Sample 3 still shows broad responses on the clamp, and sample 5 retains responses on nearby structures.

Connection to shared conditional modeling. These maps are computed from the RAW+LL score. The two branches keep separate feature groups but share coupling weights. Branch embeddings condition the scale and shift predictions on the input source. HF affects the shared weights during training, but its score is not included in the visualization. Some examples show more response on equipment, while others retain background responses. The maps describe the behavior of the trained model. Section 4.4 examines the individual design choices through ablations.

E.2 Object and Texture Responses on MVTec AD

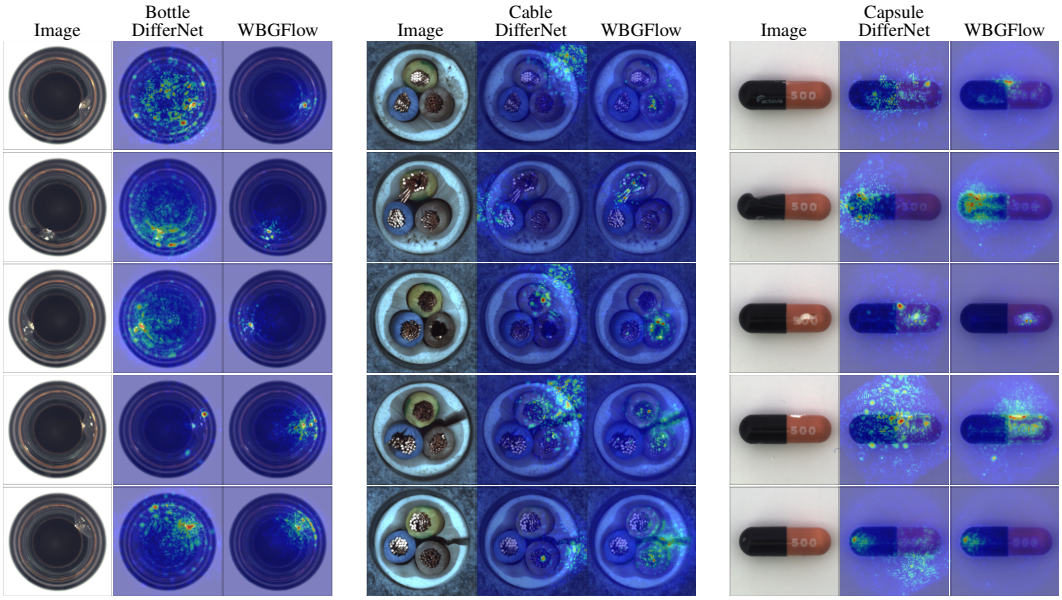

Figure E.3: MVTec AD: Bottle, Cable, and Capsule. Within each category, columns show Image, DifferNet, and WBGFlow; rows index examples from top to bottom.

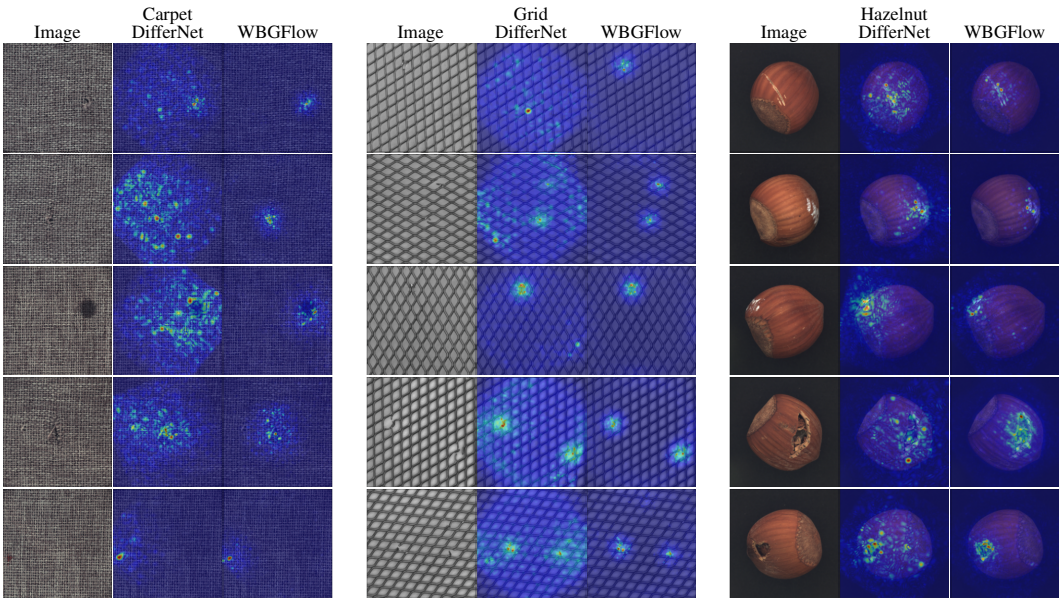

Figure E.4: MVTec AD: Carpet, Grid, and Hazelnut. The column order and row indexing follow Figure E.3.

In the first three Bottle rows, WBGFlow responds more to rim damage and less to the bottle interior. Cable rows 2 and 3 retain responses near irregular conductors. Several Capsule examples show fewer responses outside the object. Carpet and Grid maps concentrate more on local surface irregularities. Hazelnut maps often respond near visible surface damage. These changes vary across samples.

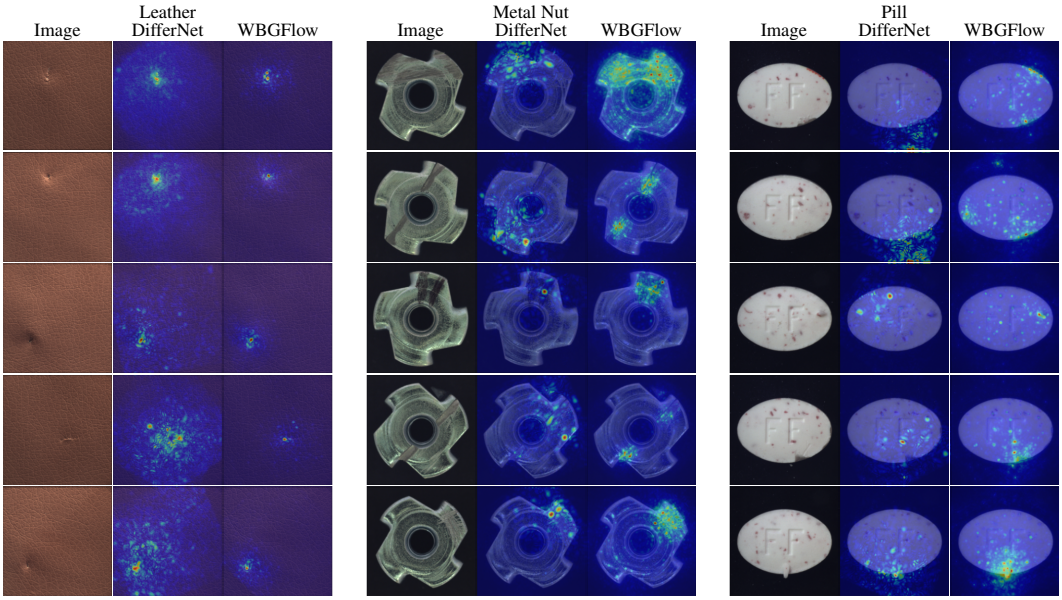

Figure E.5: MVTec AD: Leather, Metal Nut, and Pill. Each row compares the same input with the two models' saliency maps, using the established column order.

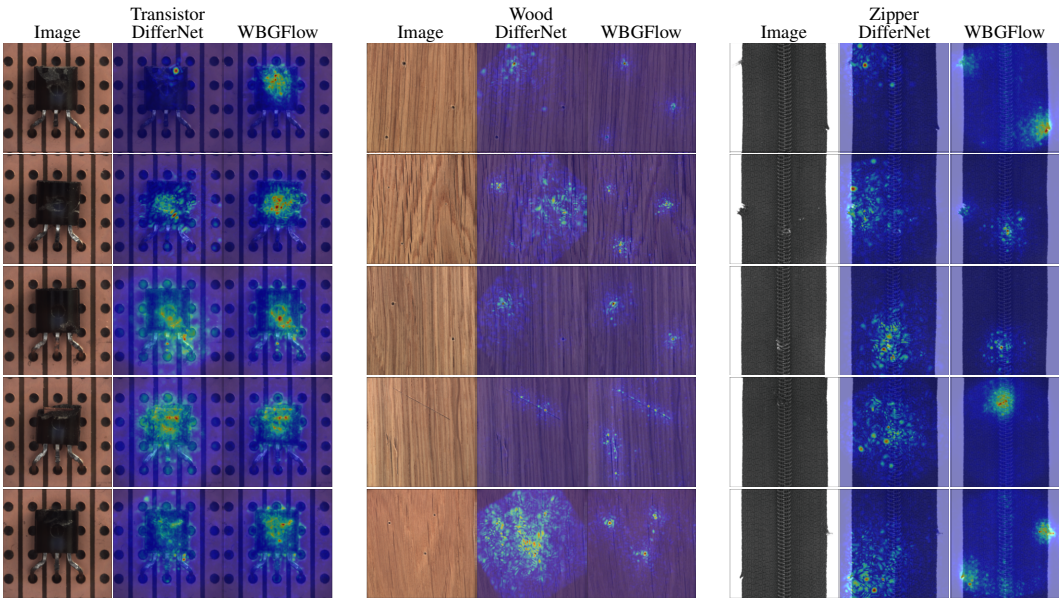

Figure E.6: MVTec AD: Transistor, Wood, and Zipper. Each category contains five examples in the same Image–DifferNet–WBGFlow column order.

Leather maps respond mainly near local surface irregularities. Metal Nut maps respond to marked or scratched areas. For Pill, more response appears on the tablet surface and less in the dark surroundings.

Transistor maps respond mainly to the component body and terminals, with some responses on the support. Wood maps show less spread across the texture and retain responses near holes or the diagonal mark in row 4. Several Zipper maps respond to fabric edges as well as the teeth. These examples show that different regions can affect the image-level score. A more concentrated map alone does not establish higher AUROC.

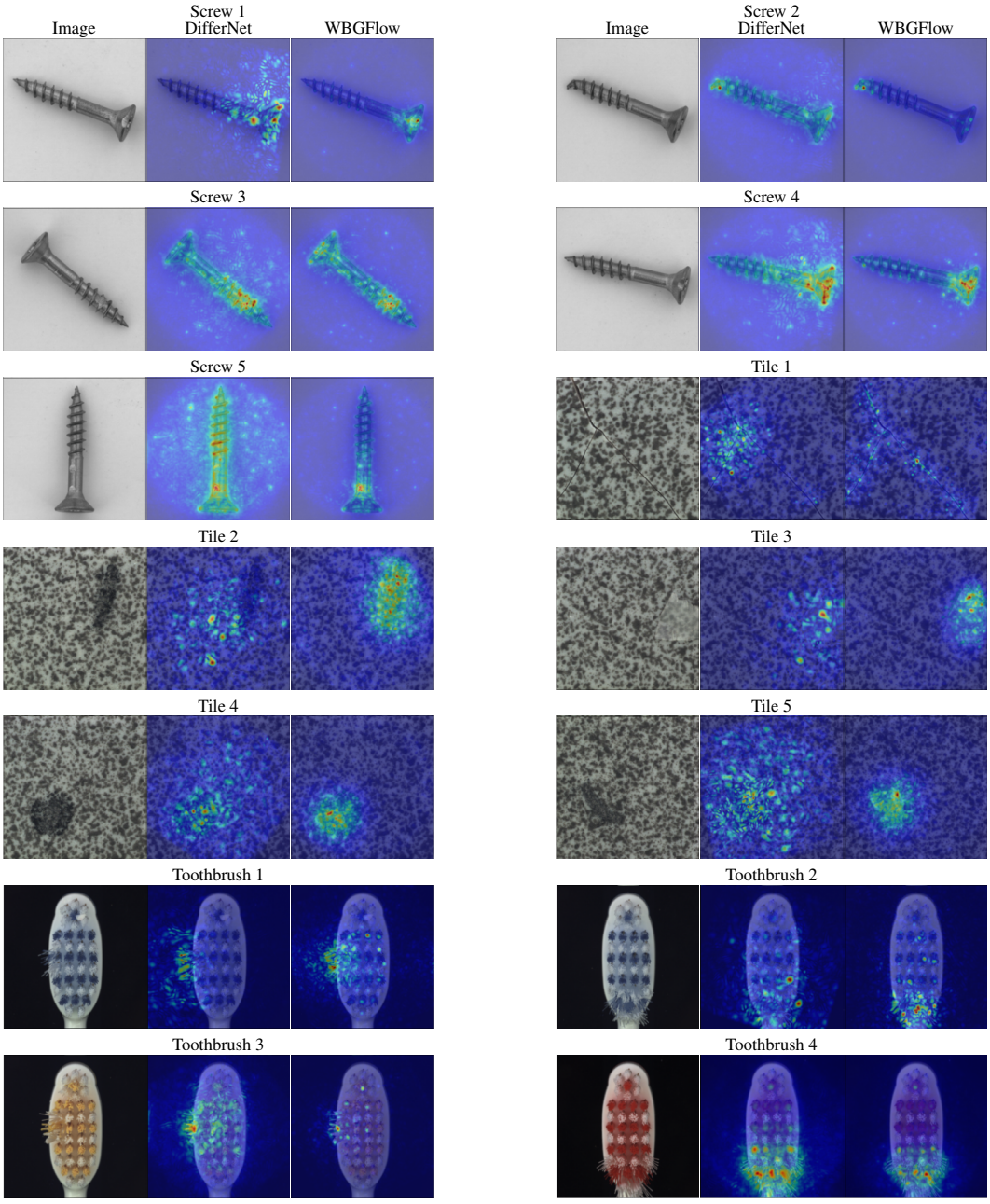

Figure E.7: MVTec AD: Screw 1–5, Tile 1–5, and Toothbrush 1–4. Each labelled triplet shows Image, DifferNet, and WBGFlow. All 14 examples are retained in an aligned grid.

Several Screw examples retain responses on the head or threads, with less spread into the surroundings. Tile and Toothbrush show mixed patterns. Some responses are localized, while others remain broad. The size of the highlighted area alone does not determine detection accuracy.