Supplementary material for Equivariant Unsupervised Object Detection with Learnable Riesz Transform and Composite Spatial Transformers

Sayan Chaki[1,2] Thierry Fournel[1] Rémi Emonet[1,2,3]

[1]Université Jean Monnet Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023 St-Etienne, France

[2]Inria, [3]Institut Universitaire de France (IUF)

sayan.chaki@inria.fr, fournel@univ-st-etienne.fr, remi.emonet@univ-st-etienne.fr

A Computing resources and model sizes

Computational resources: All the experiments were conducted on a system equipped with 48GB of RAM and an NVIDIA RTX A5000 GPU.

Runtime comparisons: We compare our full `SPAGMACE` model and with GNM, in terms of overall training time. The results are provided in Table 5. Our models, despite their improved performance, remain computationally efficient, being faster to train than GNM.

Parameter and FLOPs analysis: We report in Table 6 the number of parameters and FLOPs for different models. We use the python package "calflops" to get the FLOPs analysis. In terms of parameters, our `LeaRN-CompSTN+GMAIR` model seats between TARGET-VAE with discrete rotation groups of size 8 and 16. This allows for a fair comparison without arbitrarily degrading the architecture coherence by reducing some layer sizes. Compared to both of these, the model is more efficient in terms of FLOPs, while giving better clustering and reconstruction performance. The IRL-INR model has a very large number of parameters and requires more FLOPS.

Model	MTR-MNIST	MVTec Screws	MVTec D2S
`LeaRN-CompSTN+GMAIR`	512	794	1257
`GNM`	715	886	1648

Table 5.Training time (in minutes) across different datasets.

Model	**Parameters**	**FLOPs**
`LeaRN-CompSTN+GMAIR`	1.1M	16 GFLOPs
TARGET-VAE P8	0.9M	25 GFLOPs
TARGET-VAE P16	1.4M	29 GFLOPs
IRL-INR	80M	37 GFLOPs

Table 6.Parameter and FLOPS for different autoencoding models.

B Details on main building blocks

B.1 Details on `STN`

An STN transforms an input feature map $U \in \mathbb{R}^{H \times W \times C}$ (e.g., an image with height H, width W, and channels C) into an output feature map $V \in \mathbb{R}^{H' \times W' \times C}$. It consists of three components: a *localization network*, a *grid generator*, and a *sampler* [2].

NB: In our Figure 1, we merge the grid generator and sampler into a single "grid sampler" block for simplicity. We also allow the localization network to take as input different features that the ones sampled by the grid sampler. For instance, the first localization network of `CompSTN` (Figure 1) is expected to be rotation-invariant so it uses rotation-**invariant** features (obtained by group pooling after group convolutions) as input. On the other hand, the grid sampler must produce rotation-**equivariant** outputs, so it uses features before group pooling as input.

The **S**patial **T**ransformer **N**etwork (`STN`) process can be summarized as follows:

- *The localization network*, $f_{\text{loc}} : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^k$ predicts the parameters of spatial transformation. For a 2D affine transformation ($k = 6$): the parameters are $\theta = \{a_{11}, a_{12}, a_{21}, a_{22}, t_x, t_y\}$ and the transformation matrix is:

$$T_\theta = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

  This matrix operates in homogeneous coordinates to map input coordinates (x_s, y_s) to output coordinates (x_t, y_t).
- *The grid generator* computes a sampling grid by applying the inverse transformation T_θ^{-1} to output coordinates (x_t, y_t):

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = T_\theta^{-1} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \tag{5}$$

  For an affine transformation:

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} x_t \\ y_t \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) \tag{6}$$

  Coordinates are typically normalized to $[-1, 1]$ for consistency across image sizes.
- *The sampler* uses bilinear interpolation to compute image $V(x_t, y_t)$ from image U at source coordinates (x_s, y_s):

$$V(x_t, y_t) = \sum_{n=1}^{H} \sum_{m=1}^{W} U(m, n) \cdot \max(0, 1 - |x_s - m|) \cdot \max(0, 1 - |y_s - n|) \tag{7}$$

  The interpolation kernel $\max(0, 1 - |x_s - m|) \cdot \max(0, 1 - |y_s - n|)$ ensures differentiability, enabling gradient computations. This is applied independently to each channel.

An `STN` can be trained end-to-end by minimizing a task-specific loss whose gradients with respect to parameters θ are backpropagated through the sampler and grid generator to the localization network. This enables an `STN` to learn a geometric transformation optimizing the current task. The transformation can also be parametrized in a more constrained manner, e.g. to handle just rotation (second `STN` is `CompSTN`, in Figure 1):

$$T_\theta = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

with the rotation angle α as parameter.

The transformation can be inverted and applied in the reverse direction as done in Figure 1 ("reverse sampler"), to warp a canonical template back to the input image space.

B.2 Details on `ReResNet`

For details on rotation equivariant neural networks, that achieves a lifting operation (going into a bigger representation space but which make equivariance easier) followed by group convolutions (that preserve the equivariance), refer to [3].

As illustrated in Figure 1, the feature extractor is rotation equivariant (with respect to a discrete group, illustrated with 4 discrete rotations). In practice it extracts feature maps that can be grouped in 4 groups, each group corresponding to a specific rotation angle. The group pooling operation consists in taking the maximum response across the different rotation channels, thus producing rotation-invariant features.

B.3 Details on `CompSTN`

The goal of the `CompSTN` is to estimate a transformation that includes translation, scale and rotation, and use it to extract a "glimpse" in a canonical pose, of the original features to be used for further processing. The approach can be generalized to other transformation groups.

The `CompSTN` is a combination of two `STN` modules, combined a way to reduce the data requirements for training. The first `STN` estimates the scale and translation parameters. Intuitively, this first `STN` is expected to pinpoint the object based on its overall center of mass and size. To achieve this robustly and generalize from few data, the localization network of this first `STN` uses as input rotation-**invariant** features, obtained by applying group pooling (max over rotation channels) on the rotation equivariant features.

From the estimated scale and translation parameters, the input features are resampled to extract a "glimpse" that is centered and scaled to a canonical size. This resampled glimpse is then passed to the second `STN`, which will be responsible to extract the rotation parameter. The resampling is done on the original input features (before group pooling), so that the second `STN` can use rotation-**equivariant** features to estimate the rotation angle and generalize well.

The second `STN` thus estimates only the rotation parameter, using as input the resampled rotation-equivariant features. The final output of the `CompSTN` is obtained by resampling the original input features according to the full estimated transformation (scale, translation, rotation). In practice this final resampling takes features after the group pooling operation, so that the output features are rotation-invariant. This is a design choice to simplify further processing, compared to the other option of resampling the rotation-equivariant features (before group pooling). This choice is driven by three main considerations:

- having rotation-invariant features simplifies the subsequent processing, reducing the size of the representation,

- resampling discrete-rotation equivariant features with a rotated grid requires an interpolation scheme that create artifacts,
- end-to-end training makes it possible for the feature extractor to "bypass" the limitation of using only rotation-invariant features, by encoding richer rotation-related information in the features before group pooling.

B.4 Details on `LeaRN` (regularization)

We remind that **Lea**rnable **R**iesz-transform **N**etwork (`LeaRN`) consists in re-weighting the Riesz transform in the Fourier domain, i.e. for R_1 (similarly for R_2 and second order features):,

$$\forall \xi, \quad \mathcal{F}(\mathrm{LR}_1(I))(\xi) = \mathrm{w}_1(\xi) \cdot \mathcal{F}(R_1(I))(\xi) \tag{9}$$

where the weight function w_1 is parameterized as:

$$\forall \xi, \quad \mathrm{w}_1(\xi) = \sum_{k=1}^{K} w_{1k} \exp\Big(-\big|\, \|\xi\| - \mu_k \,\big|\Big) \tag{10}$$

To promote sparsity in the weights and peaked frequency responses that emphasize specific frequency bands, we introduce an L^1 regularization term over the weights: $R_1(\Theta) = \sum_{\ell,m} |w_{\ell m}|$. To ensure smoothness, we add an L^2 regularization term on the gradient of the radial frequency response. More precisely, if we define s_l such that $\mathrm{w}_{l(\xi)} = s_{l(\|\xi\|)}$, i.e. $s_{l(r)} = \sum_{k=1}^{K} w_{lk} \exp\Big(-\big|\, r - \mu_k \,\big|\Big)$ the regularization term is defined as s_ℓ: $R_2(\Theta) = \sum_\ell \|\nabla s_\ell\|_2^2$. This regularization term is closely linked to the weights w_{lk} with the main goal of promoting smoothness in the frequency responses, i.e. avoiding very peaky frequency re-weighting. In practice, we cross-validated using autoencoding on MTRS-MNIST to get a reasonable values: $\lambda_1 = 1e-3$ and $\lambda_2 = 5e-4$. These values are then used throughout all experiments.

In the experiments, the feature extractor is made of a `ReResNet` followed by two layers of `LeaRN`, each with Riesz orders 1 and 2. This means each input feature map produces 5 output feature maps, two with Riesz order 1 and three with Riesz order 2. After the second layer, the number of feature maps is again multiplied by 5.

B.5 Details on `SPAGMACE`

"Fake bounding-box" loss

We introduce a fake bounding-box loss

$$\mathcal{L}_{fakebbox} = \sum_i \sum_j \left(1 - \gamma^{ij}\right)^2 \cdot z^{ij}_{pres} \tag{11}$$

to discourage transparent detections with non-zero presence probability. Early experiments had shown that this regularization was important to improve over `GMAIR` (in particular for the clustering accuracy metrics).

C Theoretical Properties

C.1 On Aliasing and the Riesz transform

Aliasing occurs when a signal is sampled at a rate insufficient to capture its highest frequency components, causing high frequencies to be misrepresented as lower frequencies [42]. In CNNs, aliasing arises during downsampling operations (e.g., max-pooling or strided convolutions), which reduce the spatial resolution of feature maps. This violates

the Nyquist-Shannon sampling theorem, which states that a signal must be sampled at least twice its highest frequency to be reconstructed accurately. CNNs rely on **translation equivariance**, meaning that if an input image is shifted (translated), the output feature maps shift accordingly without changing their structure. Aliasing disrupts this property by introducing distortions, particularly incorrect phase shifts in the frequency domain, leading to inconsistent feature representations. This affects the network's ability to generalize across translated inputs, critical for tasks like image recognition [43], [44], [45].

In practice, digital processing is applied on images discretized on a grid, here assumed to be $M \times M$, and the **Discrete Fourier Transform** (DFT) is defined in the discretized frequency domain as

$$F[k,l] = \sum_{m,n=0}^{M-1} f[m,n] e^{-2\pi i(mk+nl)/M}, \quad k,l \in \{0,...,M-1\}. \tag{12}$$

to compute the frequency components of digital image $f[m,n] = f(\frac{m}{M}, \frac{n}{M}), \quad m,n \in \{0,...,M-1\}.$). Each index pair (k,l) corresponds to a frequency $\xi = (k/M, l/M)$. The discretization of the frequency domain involves a periodization of the image in the spatial domain, and the highest frequency that can be represented is $f_s = M/2$ (half the Nyquist Frequency): any frequency beyond this limit is aliased in $[-M/2, M/2]^2$.

C.1.1 Aliasing Due to Downsampling

Downsampling by a factor of 2 reduces the grid from $M \times M$ to $M/2 \times M/2$ by assuming here that M is an even integer. That typically occurs in CNNs in stride-2 convolution or pooling, effectively halving the spatial resolution. This is equivalent to reducing the sampling rate, which lowers the Nyquist frequency to ${f_{NS}}' = M/2$. The DFT of downsampled image $f_{\text{down}}[m,n] = f[2m,2n], \quad m,n \in \{0,...,(M/2)-1\}$, is:

$$F_{\text{down}}[k,l] = \sum_{m,n=0}^{M/2-1} f[2m,2n] e^{-2\pi i(mk+nl)/(M/2)}, \quad k,l \in \{0,...,(M/2)-1\}. \tag{13}$$

To understand aliasing, we need to relate F_{down} to the original DFT F. The DFT sums over the reduced grid $m,n \in \{0,...,(M/2)-1\}$. The downsampled DFT can be rewritten using the original image's DFT. The well-known standard aliasing formula gives:

$$F_{\text{down}}[k,l] = \frac{1}{4}(F[k,l] + F[k+M/2,l] + F[k,l+M/2] + F[k+M/2,l+M/2]). \tag{14}$$

The factor $1/4$ arises because the DFT of a downsampled signal averages contributions from frequency components that are shifted by multiples of the new sampling frequency. The terms $F[k+M/2,l]$, $F[k,l+M/2]$, and $F[k+M/2,l+M/2]$ represent high-frequency components (beyond $M/4$) that "fold" into the lower frequency range $k,l \in \{0,...,M/2-1\}$.

The high-frequency components (e.g., $F[k+M/2,l]$) are outside the limit $M/4$. These components alias into the lower frequency range, causing distortions because they are indistinguishable from lower frequencies in the downsampled signal.

C.1.2 Translation and Phase Errors

Now, consider a translated image $f[m-b_0, n-b_1]$. In the frequency domain, such a translation introduces a phase shift: component $F[k,l]$ is modulated as follows: $F[k,l]e^{-2\pi i(b_0k+b_1l)/M}$, and the DFT after downsampling becomes $\frac{1}{4}\sum_{p,q\in\{0,M/2\}} F[k+p, l+q]e^{-2\pi i(b_0k+b_1l)/M}e^{-2\pi i(b_0p+b_1q)/M}$.. For aliased terms (e.g., $p = M/2$), the modulation term expresses as $e^{-2\pi ib_0(k+M/2)/M} = e^{-2\pi ib_0k/M}e^{-\pi ib_0}$. where $e^{-\pi ib_0}$ is an additional factor due to alisasing, causing a phase error which disrupts translation equivariance. The downsampled image's frequency components are modulated incorrectly, meaning the network's response to a translated input is not a simple shift of the original response. Non-linearities (e.g., ReLU) worsen this by generating higher harmonics, increasing the impact of aliased frequencies.

C.1.3 The Riesz Transform Solution

The Riesz transform provides a steerable, multi-directional representation. This allows for selective filtering of high-frequency components in specific directions, which can help design anti-aliasing filters that preserve important features while suppressing aliasing artifacts [46], [47].

Proof that Riesz resolves aliasing in translation and rotation equivariance: Our scale-adaptive transform enables adaptive sampling and filtering that prioritises critical signal components, reducing aliasing in regions with rapid change.

Since $\widehat{\mathcal{R}_j f}(\xi) = i\frac{\xi_j}{|\xi|}\hat{f}(\xi)$, translating f by b gives:

$$\mathcal{R}_j(\widehat{f(\cdot - b)})(\xi) = e^{-2\pi i\xi\cdot b}\widehat{\mathcal{R}_j f}(\xi) \tag{15}$$

The factor $\frac{\xi_j}{|\xi|}$ is independent of b, so the phase shift factors out exactly. In the spatial domain this reads $\big(\mathcal{R}_j f\big)[m-b_0, n-b_1]$ — the feature map shifts with the input, giving full translation equivariance at native resolution.

Define $g_{j[m,n]} = \big(\mathcal{R}_j f\big)[m,n]$. The downsampled signal by factor 2 and its DFT are:

$$g_{j,\text{down}}[m,n] = g_{j[2m,2n]}, \quad m,n \in \left\{0, ..., \frac{M}{2} - 1\right\} \tag{16}$$

$$G_{j,\text{down}}[k,l] = \sum_{m,n=0}^{\frac{M}{2}-1} g_{j[2m,2n]}e^{-2\pi i(mk+nl)/(M/2)} \tag{17}$$

By the standard DFT aliasing identity:

$$G_{j,\text{down}}[k,l] = \frac{1}{4}\sum_{(p,q)\in\{0,\frac{M}{2}\}^2} G_{j[k+p,l+q]} \tag{18}$$

For the translated signal each term expands as:

$$G_{j[k+p,l+q]} = i\frac{k+p}{\sqrt{(k+p)^2+(l+q)^2}}F[k+p,l+q]e^{-2\pi i(b_0(k+p)+b_1(l+q))/M} \tag{19}$$

Factoring the phase, the exponential splits cleanly into a primary shift and an aliasing residual:

$$e^{-2\pi i(b_0(k+p)+b_1(l+q))/M} = \underbrace{e^{-2\pi i(b_0k+b_1l)/M}}_{\text{primary shift}} \cdot \underbrace{e^{-2\pi i(b_0p+b_1q)/M}}_{\text{aliasing phase}} \tag{20}$$

Pulling the primary shift outside the sum:

$$G_{j,\text{down}}[k,l] = \frac{1}{4}e^{-2\pi i(b_0k+b_1l)/M} \times \sum_{(p,q)\in\{0,\frac{M}{2}\}^2} \left[i\frac{k+p}{\sqrt{(k+p)^2+(l+q)^2}} F[k+p, l+q] e^{-2\pi i(b_0p+b_1q)/M} \right] \tag{21}$$

The aliased terms ($p = \frac{M}{2}$ or $q = \frac{M}{2}$) introduce residual phases such as $e^{-\pi i b_0}$, which modulate the amplitude of high-frequency components $F\left[k + \frac{M}{2}, l\right]$.

The aliasing phases do **not** disturb the primary translation phase $e^{-2\pi i(b_0k+b_1l)/M}$. The Riesz directional filter $\frac{k+p}{\sqrt{(k+p)^2+(l+q)^2}}$ encodes directional structure at full resolution, and since the phase shift is applied **before** downsampling, aliasing affects only the mixing of frequency amplitudes, not translation equivariance.

In the spatial domain, the downsampled feature map is:

$$g_{j,\text{ down}}[m,n] = g_j[2m,2n] = \big(\mathcal{R}_j f\big)[2m,2n]. \tag{22}$$

For the translated signal:

$$g_{j,\text{ down}}[m,n] = \big(\mathcal{R}_j f\big)[2m-b_0, 2n-b_1]. \tag{23}$$

The downsampled feature map of the translated signal is a shifted version of the original downsampled feature map, preserving the translation structure. The aliased terms in the frequency domain affect the magnitude of the features (due to $e^{-\pi i b_0}$) but do not disrupt the phase shift, ensuring equivariance.

The Riesz transform's output G_j represents directional features that are inherently equivariant to translation. Downsampling introduces aliasing, but because the phase shift is encoded at full resolution, the aliased terms only contribute to the feature map's amplitude, not its phase structure. The directional filter ensures that the features transform predictably, even after aliasing.

The Riesz transform is computed at full resolution ($M \times M$), capturing all frequencies up to $M/2$. This prevents information loss before downsampling, unlike standard CNNs, where downsampling occurs directly on the signal or convolved features.

D Riesz Enhanced Equivariant Networks Generalize better

Generalizability: Given an equivariant network incorporating steerable group convolutions, we propose augmenting the network with a learnable steerable Riesz transform upstream to the initial group convolution layers in the aim to build a more generalizable architecture able to well approximate equivariant feature maps ϕ (see definition below). Theorem below, about Riesz Equivariant feature bounds confirms generalizability on the basis of homogeneous bounds for equivariant networks.

Theorem (Homogeneous Bounds for Equivariant Networks) [48]
For any equivariant network f, with high probability:

$$\mathcal{L}(f_{\boldsymbol{W}}) \leq \hat{\mathcal{L}}_\gamma(f_{\boldsymbol{W}}) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\prod_l \|\boldsymbol{W}_l\|_2^2}{\gamma^2 m\eta} \cdot \left(\sum_{l=1}^{L} \sqrt{M(l,\eta)}\right)^2 \cdot \sum_l \frac{\sum_{\psi,i,j} \left\|\widehat{\boldsymbol{W}}_l(\psi,i,j)\right\|_F^2 / \dim_\psi}{\|\boldsymbol{W}_l\|_2^2}}\right) \tag{24}$$

where $\mathcal{L}(f_W)$ represents the true expected loss and $\hat{\mathcal{L}}_\gamma(f_W)$ is the empirical margin loss. γ denotes the classification margin parameter, while $\eta \in (0,1)$ is a perturbation probability parameter. L is the number of network layers, with W_l representing the weight matrices at layer l. $B = \max\left(1, \prod_l \|W_l\|_2\right)$ bounds the network output, and $\beta = \prod_l \|W_l\|_2$ is the product of spectral norms. $\widehat{W}_l(\psi,i,j)$ are the kernel parameters in the Fourier domain for irreducible representation ψ, with $\dim_\psi$ being the dimension of that representation and $M(l,\eta) := \log\left(\frac{\sum_{l=1}^{L}\sum_\psi m_{l,\psi}}{1-\eta}\right) \max_\psi\left(5 m_{l-1,\psi} m_{l,\psi} c_\psi\right)$.

An ideal equivariant feature map $\phi : L^2(\mathbb{R}^2, \mathbb{R}^{c_{\text{in}}}) \to L^2(\mathrm{SO}(2), \mathbb{R}^{c_{\text{out}}})$ satisfies the following property: for any $x \in L^2(\mathbb{R}^2, \mathbb{R}^{c_{\text{in}}})$ and $h \in \mathrm{SO}(2)$, with the action $x_h(y) = x(h^{-1}y)$ for $y \in \mathbb{R}^2$, the feature map is equivariant under $\mathrm{SO}(2)$, i.e.,

$$\phi(x_h)(g) = \phi(x)(hg) \tag{25}$$

for all $g \in \mathrm{SO}(2)$.

Assumption 1 (Spectral Misalignment). Let $\tilde{x}$ and $\tilde{\mathcal{R}}(x)$ denote the lifts of x and its Riesz transform to $L^2(SO(2))$. We say x satisfies the Spectral Misalignment Assumption if:

$$\sum_i \sum_{|m|>1} |\hat{\tilde{x}}(m)_i|^2 > \sum_i \sum_{|m|>1} |\widehat{\tilde{\mathcal{R}}(x)}(m)_i|^2 \tag{26}$$

$\tilde{x}$ has strictly more energy than $\tilde{\mathcal{R}}(x)$ at $SO(2)$-Fourier modes $|m| \neq 1$.

Assumption 2 (Joint Spectral Concentration): Let $F_l = \mathcal{F}(\text{input to layer } l)$ be expressed in polar coordinates (r,θ), with angular Fourier coefficients $c_m^{(l)}(r)$. We say the learned radial weight $w_l : \mathbb{R}_+ \to \mathbb{R}_+$ satisfies the **Joint Spectral Concentration** condition if both of the following hold:

High-mode non-amplification. For all $|m| > 1$ and a.e. $r > 0$:

$$w_{l(r)} \leq 1 \quad \text{on the support of} \quad \left\{r : c_m^{(l)}(r) \neq 0\right\}. \tag{27}$$

Low-mode error reduction. There exists a measurable set $S \subset \mathbb{R}_+$ of positive measure such that, for $m \in \{pm1\}$:

$$\int_0^\infty w_{l(r)}^2 \left|c_{pm1}^{(l)}(r) - \hat{\varphi}_{pm1}(r)\right|^2 r\,\mathrm{d}r \int_0^\infty \left|c_{pm1}^{(l)}(r) - \hat{\varphi}_{pm1}(r)\right|^2 r\,\mathrm{d}r. \tag{28}$$

$\tilde{x}$ has strictly more energy than $\tilde{\mathcal{R}}(x)$ at $SO(2)$-Fourier modes $|m| \neq 1$.

Interpretation. (26) prevents w_l from amplifying the spurious high-angular-mode energy introduced by upstream nonlinearities. (27) requires that w_l strictly improves the radial approximation of the target filter at the modes $m = pm1$ that matter for equivariance. These conditions are jointly sufficient to guarantee $\delta > 0$ below; neither alone suffices.

Theorem (Riesz Equivariant Feature Bounds). Let $x \in L^2(\mathbb{R}^2)$ and let $\varphi(x) \in L^2(SO(2), \mathbb{R}^{c_{\text{out}}})$ be an ideal $SO(2)$-equivariant feature map satisfying Assumption 1 (i.e. $\hat{\varphi}(m)_i \approx 0$ for $|m| > 1$). Under the Spectral Misalignment Assumption, the lift $\tilde{\mathcal{R}}(x)$ of the Riesz transform and the $\widetilde{LR}(x)$ the SO(2)-lift of the LeaRN output satisfies:

$$\left\| \|\widetilde{LR}(x) - \varphi(x)\| \right\|_2 \leq \left\| \tilde{\mathcal{R}}(x) - \varphi(x) \right\|_2 < \|\tilde{x} - \varphi(x)\|_2 \tag{29}$$

> The Riesz-lifted feature is a strictly better approximation to any ideal equivariant target than the raw lift $\tilde{x}$.

Through our proof, below, we argue that if the Riesz representation tightens the homogeneous bound it would in turn improve the model's generalization.

The section is devoted to the proof of the above Theorem indicating that given an equivariant network incorporating steerable group convolutions, augmenting the network with a learnable steerable Riesz transform upstream to the initial group convolution layers leads to a more generalizable architecture able to well approximate equivariant feature maps ϕ (see definition 1).

D.0.1 Why the Assumption holds for natural images?

Fourier modes $\hat{\phi}(\xi, m)_i$ are significant only for $|m| < 1$ i.e. $\hat{\phi}(\xi, m)_i \approx 0 \quad \text{for} \quad |m| \geq 1, \quad i = 1, ..., c_{\text{out}}$.

The equivariance constraint, defined as $\phi(x_h)(g) = \phi(x)(hg) \quad \text{for} \quad x_h(y) = x(h^{-1}y)$, where $h \in SO(2)$, restricts $\phi(x)$ to transform predictably under rotations. For features such as gradients, this transformation manifests as a phase shift (e.g., a rotation of direction), which corresponds to Fourier modes $m = \pm 1$. Higher $|m|$ modes are associated with more intricate transformations, such as those of higher-order tensors, which are less prevalent in typical image features due to their increased rotational complexity [49].

Group convolutional neural networks are specifically designed to exploit group symmetries, in this case, $SO(2)$. The feature map $\phi(x)$ is often conceptualized as an idealized output of a GCN layer, which applies filters that are equivariant under $SO(2)$. These filters typically generate low-m features, as higher-m modes necessitate more complex kernels, which are less common in the early layers of such networks. Consequently, Assumption 1 is naturally satisfied when $\phi(x)$ represents the output of such layers, capturing simple rotational patterns like edges or oriented textures.

In the context of natural images, such as photographs, the presence of edges and textures supports the validity of Assumption 1. For instance, the outline of a tree or a building rotates as a vector field, aligning with $m = \pm 1$ in the $SO(2)$-Fourier domain. These low-m modes effectively approximate the dominant features in such images, reinforcing the assumption that $\hat{\phi}$ is concentrated at $|m| < 1$ [50].

However, Assumption 1 may not hold if $\phi(x)$ captures high-order symmetries, such as textures with rapid rotational variation. For example, starfish-like patterns exhibit complex rotational patterns that require $|m| \geq 1$. In such cases, the Riesz transform's concentration at $m = \pm 1$ may result in a larger distance $\| R(x) - \phi(x) \|_2$, potentially

violating the inequality. This highlights the importance of aligning the feature map's spectral properties with the task and data at hand.

In summary, Assumption 1 generally holds when $\phi(x)$ is designed to capture low-order, rotation-equivariant features, such as gradients or edges, which are prevalent in natural images and early GCN layers. Its validity is less certain for intricate, high-frequency patterns, necessitating careful consideration of the feature map's design in applications where the inequality is applied.

D.1 Riesz as a pre-processing step for equivariant networks

Consider two classifier architectures:

$$\begin{aligned} \text{Standard:} \quad & f_1(x) = V(W_1 * x) \\ \text{Riesz-augmented:} \quad & f_2(x) = V(W_2 * R(x)) \end{aligned} \tag{30}$$

where W_1, W_2 are convolutional filters, $R(\cdot)$ is the Riesz transform, and V is a shared linear classifier that maps features to class logits.

For both architectures to perform equivalently, the convolutional layers must approximate the ideal feature map such that $W_1 * x \approx \phi(x)$ and $W_2 * R(x) \approx \phi(x)$. Our key insight is that the Riesz transform aligns the input representation with the symmetry structure of $\phi(x)$, reducing the complexity required in the subsequent convolutional layer.

Formally, we analyze the minimal-norm solutions,

$$W_1^* = argmin_{W \in \mathcal{W}} \|W\|_F, W_2^* = argmin_{W \in \mathcal{W}} \|W\|_F \tag{31}$$

If $\mathcal{R}(x)$ reduces the "distance" to $\phi(x)$, W_2 requires less magnitude what directly translates to tighter PAC-Bayesian generalization bounds, as evidenced by the coefficient $\sum_l \sum_{\psi,i,j} \frac{\|\widehat{W}_l(\psi,i,j)\|_F^2}{\dim \psi} \|W_l\|_2^2$. This distance is effectively reduced, as shown by theorem 2 formulated and proved below:

Here, we have finally established that enhancing the convolution layers, with our learnable steerable Riesz network has significant advantages: (a) It prevents aliasing (b) It encapsulates object features better (c) It makes existing equivariant networks generalize better.

This scaling allows for better visualization and analysis of fine-grained details, which is essential for capturing object features where objects are defined by strong edges. Moreover, scaled features improve numerical stability when these representations are used in downstream algorithms, such as neural networks, by preventing large magnitude differences from skewing results. Ultimately, scaling enhances the network's ability to leverage equivariant features, leading to more robust and effective performance in frequency-domain tasks.

Proof of Riesz Equivariant Feature Bounds: We expand the squared norm via the SO(2) Fourier–Parseval identity:

$$\|f\|_2^2 = 2\pi \sum_i \sum_{m \in \mathbb{Z}} |\hat{f}(m)_i|^2 \tag{32}$$

The lift of the Riesz transform to $SO(2)$ via projection onto $(\cos\alpha, \sin\alpha)$ takes the exact form

$$\tilde{\mathcal{R}}(x)_i(\alpha) = a_i e^{i\alpha} + b_i e^{-i\alpha} \tag{33}$$

The $SO(2)$-Fourier coefficients of $\tilde{\mathcal{R}}(x)$ vanish at every frequency $|m| \neq 1$. This is an exact algebraic consequence of the gradient-like directional structure of the Riesz kernel.

Under Assumption 1, Parseval decomposes the approximation error by frequency regime:

$$\|f - \varphi(x)\|_2^2 = 2\pi \sum_i \left[\underbrace{\sum_{|m| \leq 1} |\hat{f}(m)_i - \hat{\varphi}(m)_i|^2}_{\text{low-frequency terms}} + \underbrace{\sum_{|m|>1} |\hat{f}(m)_i|^2}_{\text{high-frequency residual}} \right] \tag{34}$$

At modes $|m| > 1$: (41) gives $\tilde{\mathcal{R}}(x)$ zero contribution, while $\tilde{x}$ contributes positively by Assumption A. Define the **spectral gap**

$$\Delta := 2\pi \sum_i \sum_{|m|>1} |\hat{\tilde{x}}(m)_i|^2 > 0 \tag{35}$$

Let $\varepsilon_{|m|=1}$ bound the worst-case disadvantage of $\tilde{\mathcal{R}}(x)$ at low frequencies. Assumption A ensures $\Delta > \varepsilon_{|m|=1}$.

Combining the above estimates:

$$\|\tilde{x} - \varphi(x)\|_2^2 - \left\|\tilde{\mathcal{R}}(x) - \varphi(x)\right\|_2^2 \geq \Delta - \varepsilon_{|m|=1} > 0 \tag{36}$$

The high-frequency gain Δ strictly outweighs any low-frequency loss, establishing the result.

Exntended prood for LeaRN Equivariant Feature Approximation Bounds: We extend the inequality for LeaRN and prove that,

$$\left\|\widetilde{LR}(x) - \varphi(x)\right\|_2 < \left\|\tilde{\mathcal{R}}(x) - \varphi(x)\right\|_2, \tag{37}$$

where $\widetilde{LR}(x)$ denotes the SO(2)-lift of the LeaRN output, $\tilde{\mathcal{R}}(x)$ the lift of the plain Riesz baseline, and $\tilde{x}$ the lift of the raw input.

The LeaRN learned Riesz component applies a radial weight $w_1(r)$ to the Riesz kernel output:

$$\mathcal{F}(LR_1(I))(r,\theta) = w_1(r) \cdot (-i\cos\theta)\hat{I}(r,\theta) = w_1(r) \sum_{m\in\mathbb{Z}} c_m^{R(r)} e^{im\theta}. \tag{38}$$

By Assumption 2, $w_1(r) \leq 1$ on the support of all high-mode content $|m| > 1$. Consequently, the high-mode energy in the LeaRN output satisfies

$$\sum_{|m|>1} \int_0^\infty w_1(r)^2 \, |c_m^{R(r)}|^2 \, r \, \mathrm{d}r \leq \sum_{|m|>1} \int_0^\infty |c_m^{R(r)}|^2 \, r \, \mathrm{d}r. \tag{39}$$

This replaces the erroneous claim of zero high-mode support (which holds only for purely scalar inputs) with the rigorously valid non-amplification bound By Parseval's theorem on $L^2(\mathbb{R}^2)$ and the unitarity of the SO(2)-Fourier series, the full squared approximation error of the LeaRN output decomposes as:

$$\left\|\widetilde{LR}(x) - \varphi(x)\right\|_2^2 = 2\pi \sum_{m\in\mathbb{Z}} \int_0^\infty |w_1(r) c_m^{R(r)} - \hat{\varphi}_{m(r)}|^2 \, r \, \mathrm{d}r. \tag{40}$$

Using Assumption 1 ($\hat{\varphi}_m = 0$ for $|m| > 1$), we split into low-mode and high-mode contributions:

$$\left\|\widetilde{LR}(x)-\varphi(x)\right\|_2^2 =$$

$$\underbrace{2\pi\sum_{|m|\le 1}\int_0^\infty |w_1(r)c_m^{R(r)}-\hat{\varphi}_{m(r)}|^2\ r\,\mathrm{d}r}_{A_{\text{low}}}+\underbrace{2\pi\sum_{|m|>1}\int_0^\infty w_1(r)^2\ |c_m^{R(r)}|^2\ r\,\mathrm{d}r}_{A_{\text{high}}}. \tag{41}$$

The analogous decomposition for the plain Riesz baseline (i.e., $w_1 \equiv 1$) gives:

$$\left\|\tilde{\mathcal{R}}(x)-\varphi(x)\right\|_2^2 =$$

$$\underbrace{2\pi\sum_{|m|\le 1}\int_0^\infty |c_m^{R(r)}-\hat{\varphi}_{m(r)}|^2\ r\,\mathrm{d}r}_{B_{\text{low}}}+\underbrace{2\pi\sum_{|m|>1}\int_0^\infty |c_m^{R(r)}|^2\ r\,\mathrm{d}r}_{B_{\text{high}}}. \tag{42}$$

We bound each pair of terms separately.

High-mode terms.

$$A_{\text{high}} \le B_{\text{high}}, \quad \text{i.e.,} \quad A_{\text{high}} - B_{\text{high}} \le 0. \tag{43}$$

Low-mode terms. By Assumption 2 applied to $m = pm1$ (and the $m = 0$ terms are equal since the Riesz kernel contributes zero energy at $m = 0$ for a real input, or else JSC-ii extends to cover them,

$$A_{\text{low}} < B_{\text{low}}, \quad \text{i.e.,} \quad A_{\text{low}} - B_{\text{low}} < 0. \tag{44}$$

Adding both contributions:

$$\left\|\widetilde{LR}(x)-\varphi(x)\right\|_2^2 - \left\|\tilde{\mathcal{R}}(x)-\varphi(x)\right\|_2^2 = (A_{\text{low}} - B_{\text{low}}) + \left(A_{\text{high}} - B_{\text{high}}\right) < 0. \tag{45}$$

We therefore define the **spectral improvement gap**

$$\delta := (B_{\text{low}} - A_{\text{low}}) + \left(B_{\text{high}} - A_{\text{high}}\right) > 0, \tag{46}$$

which is strictly positive by Assumption 2, and additionally non-negative in the high-mode contribution. We have established:

$$\left\|\widetilde{LR}(x)-\varphi(x)\right\|_2^2 = \left\|\tilde{\mathcal{R}}(x)-\varphi(x)\right\|_2^2 - \delta < \left\|\tilde{\mathcal{R}}(x)-\varphi(x)\right\|_2^2. \tag{47}$$

The inequality is non-strict, holding with equality only in the degenerate case that the raw input is already mode-concentrated. This completes the proof.

E Details on experimental results

E.1 Ablation study and extended results on SR-MNIST

We report and detailed exhaustive results of our ablation study on the SR-MNIST dataset in Table 7. We also compare to state of the art models: TARGET-VAE (with two discrete rotation group sizes), CODAE and IRL-INR. Our method `LeaRN-CompSTN`+GMVAE outperforms the other methods on the three metrics, with both low data (80% of the total) and the full training set.

Our `CompSTN` gives better results that the approach used by TARGET-VAE (which differs also in the way the decoding is performed, using a neural approximation instead of a deconvolutional neural network). This holds whether we use TARGET-VAE with

	Low Data			More Data		
	NMI	ARI	SSIM	NMI	ARI	SSIM
SR-MNIST						
`STN`+VAE	0.59	0.51	0.75	0.65	0.57	0.78
`STN`+GMVAE	0.6	0.52	0.75	0.69	0.61	0.79
`CompSTN`+GMVAE	0.68	0.59	0.81	0.81	0.70	0.92
CODAE [15]	0.62	0.52	0.76	0.76	0.66	0.83
IRL-INR [13]	0.65	0.51	0.69	0.83	0.78	0.94
TARGET-VAE P_8 [12]	0.63	0.51	0.77	0.78	0.65	0.86
TARGET-VAE P_{16} [12]	0.65	0.52	0.8	0.78	0.67	0.88
`LeaRN`+TARGET-VAE P_{16}	0.69	0.60	0.82	0.82	0.72	0.93
`LeaRN-CompSTN`+GMVAE	**0.78**	**0.62**	**0.83**	**0.88**	**0.80**	**0.95**

Table 7. Ablation and comparison on clustering and reconstruction metrics on SR-MNIST (auto-encoding scaled and rotated MNIST) dataset. (extension of Table 3)

Model	**Accuracy (%)**
GroupConv (GCNN)	82.76
Standard CNN	70.41
Riesz + CNN	89.78
Riesz + GCNN	93.45
`LeaRN` + GCNN	**98.44**
Standard DFT + CNN	73.47
Standard DFT + GCNN	83.58

Table 8. Classification accuracy on rotated-MNIST dataset. (reproduction of Table 4)

discrete rotation groups of size 8 or 16, or even our improved version with `LeaRN`. Our `LeaRN` also improves even used as a feature extractor for TARGET-VAE.

Strarting from a simple baseline that uses `STN` and a standard VAE, we observe a minor improvement by using a GMVAE (Gaussian Mixture VAE) and a greater improvements when using `LeaRN` instead of `STN`.

.

E.2 Comparison of feature extractor Table 4

To compare the our `LeaRN` feature extractor in a simpler setting (simpler than unsupervised learning), we evaluate classification accuracy on the rotated-MNIST dataset (see Table 8).

The best performance is achieved by using `LeaRN` in conjunction with GCNN (Group-Equivariant Neural Networks). Around 5 points of accuracy improvement is observed compared to using an *unlearned* Riesz transform: our method that learns weights is thus effective. Even an unlearned Riesz transform outperforms other baselines that use plain CNN or GCNN. Using a standard Fourier transform improves over working directly in the spatial domain but is still outperformed by `LeaRN` or an unlearned Riesz transform.

E.3 Visuals for results

We provide additional qualitative results that could not be added in the main paper due to page limit constraints. See figure captions for details.

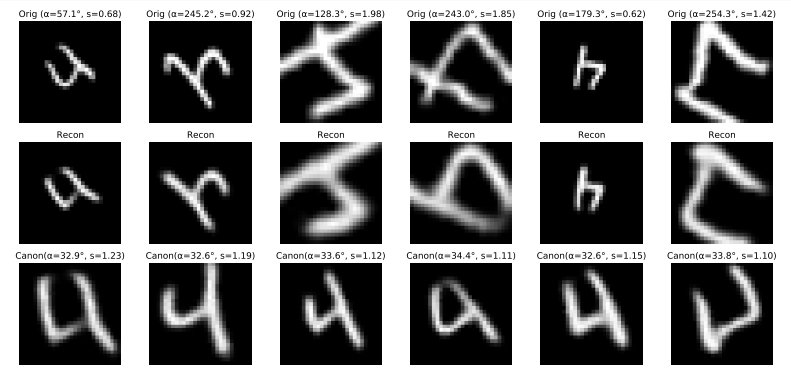

Figure 4. The canonicalization of the digit 4. The first row is the original rotated and scaled digit, the second row is the reconstructed image, and the final row is the reconstruction canonicalized. All results are produced by our `LeaRN-CompSTN`+GMVAE. While not vertical (as there is no supervision signal to guide the recovery of the angle, the model captures it at best with a systematic offset), the model is able to properly rotate and scale the digits.

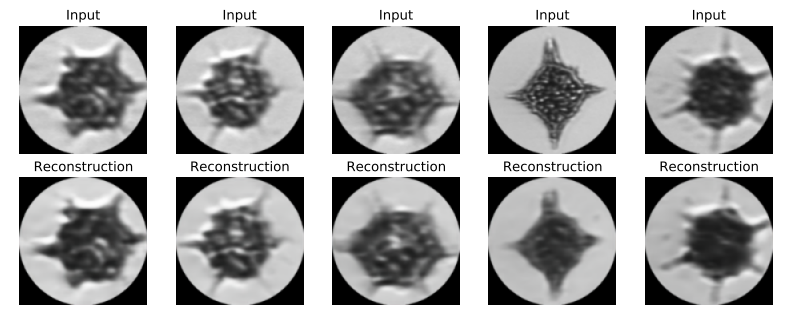

Figure 5. Qualitative visualisation of the WHO-Plankton dataset. The first row are the original samples, and the second row corresponds to the reconstructions by our `LeaRN-CompSTN`+GMVAE.

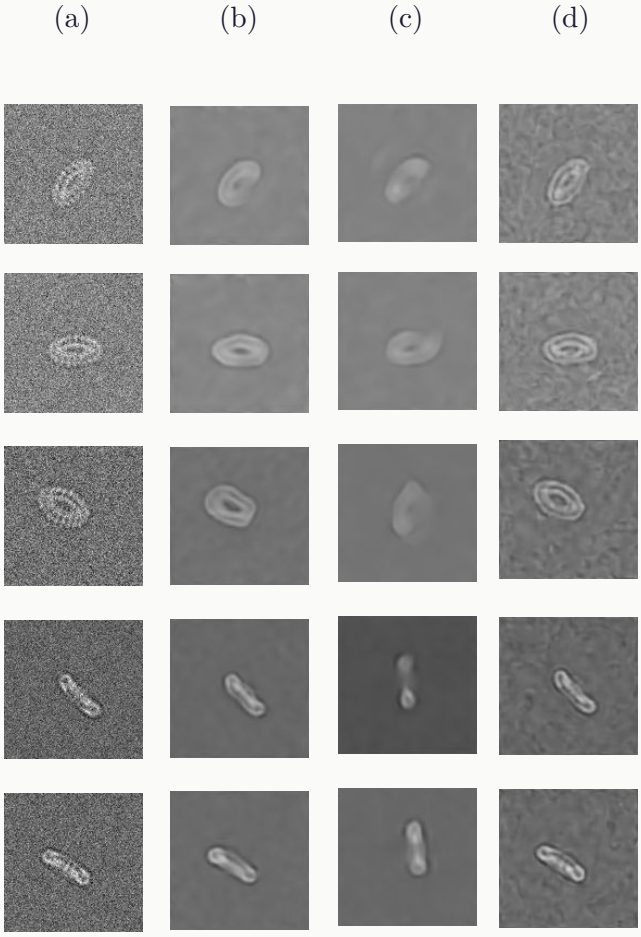

Table 9. The reconstruction results from the benchmark models on Tomotwin Cryo-EM dataset. (a) Original images, (b) IRL-INR, (c) TARGET-VAE, (d) `LeaRN-CompSTN`+GMVAE. `LeaRN-CompSTN`+GMVAE is able to better reconstruct the images compared to other methods, keeping sharper edges/details than the best performing IRL-INR.

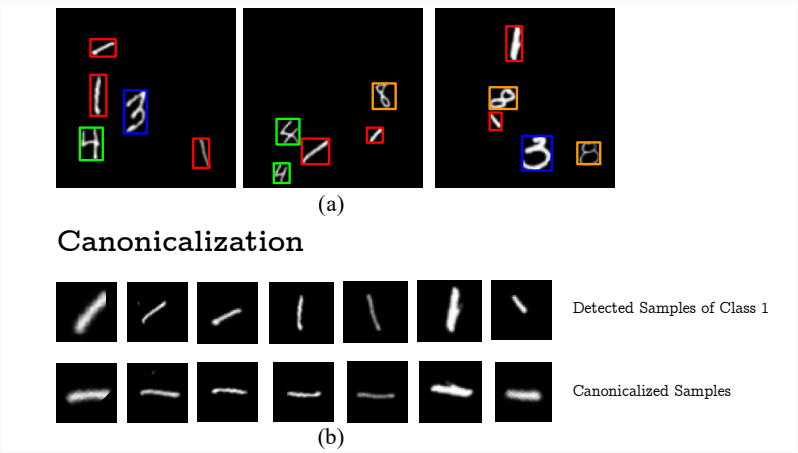

Figure 6. Qualitative visualization of the results of our model `LeaRN-CompSTN+GMAIR` on the MTRS-MNIST Dataset. The first figure (a) depicts how our model efficiently detects and clusters different class of digits in the image. The second figure (b) shows canonicalization for the digit '1' that has been detected in the 3 image samples provided in (a).

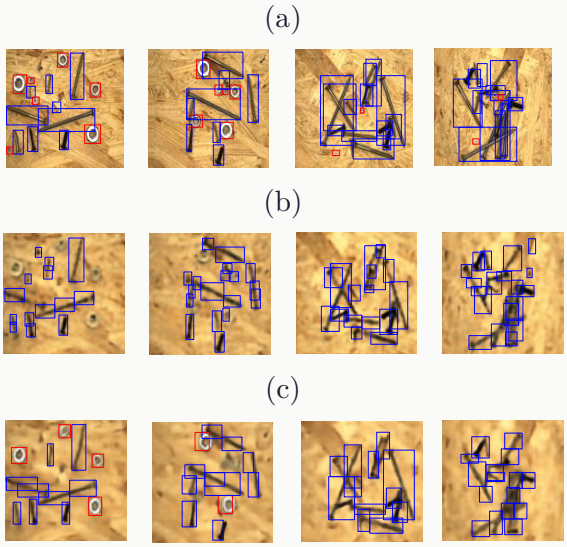

Figure 7. Qualitative visualizations of the object discovery (i.e., no boxes or labels provided) task on Low Data setting for MVTec Screws dataset. (a) Original Images with ground truth boxes (used only for evaluation), (b) GNM results (c) `LeaRN-CompSTN`+SPAGMACE results. Boxes are colored by cluster assignments (as done in the clustering metrics for the tables). While both methods tend to over segment the screws, ours is better at capturing the second type of object. and better captures different screw scales (e.g. the big screw on the right, third column).