CRE: CONVOLUTIONAL BASIS-FREE ROTATION-EQUIVARIANT OPERATOR ON GRIDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific and medical imaging often suffers from limited labeled data, making it important to exploit structure already present in the data. These images often contain geometric symmetries. In particular, rotation is a common nuisance factor: tissues, cells, and objects can appear at arbitrary orientations. Standard CNNs exploit translation symmetry but lack built-in rotation handling, while existing equivariant methods often rely on discrete rotation groups, predefined steerable bases, lifted orientation spaces, or non-standard image representations. We propose CRE, a local, basis-free convolutional block that models rotations directly on the native image grid without explicit orientation channels. CRE combines center-relative pixel interactions with invariant neighborhood attention, replacing fixed Cartesian kernel weights with geometry-conditioned local relations. We evaluate CRE on six benchmarks: three classification datasets and three segmentation datasets. In classification, CRE achieves the best mean accuracy across continuous rotations on all three datasets, while maintaining competitive variance over different angles. In segmentation, CRE remains competitive with baselines showing that it can be used as a practical simile for CNN-based pipelines. Code will be released.

1 INTRODUCTION

Modern machine learning models often require large and diverse training sets to generalize reliably (Sun et al., 2017). This requirement is especially problematic in scientific and medical imaging (Oakden-Rayner, 2020; Puyol-Antón et al., 2021; Garcia et al., 2025), where expert annotations are expensive, data acquisition is costly, and the available samples may cover only a limited range of conditions. In these domains, improving generalization by collecting more labeled data is often unrealistic (Tajbakhsh et al., 2016), but an alternative is to exploit known structure in the data itself (Cohen & Welling, 2016; Weiler & Cesa, 2019).

Many visual data domains contain geometric symmetries (Cohen & Welling, 2016; Weiler & Cesa, 2019). When such symmetries are built into the model, the network does not need to learn the same pattern independently under every possible transformation. This can improve generalization to unseen poses, and yield more predictable responses under controlled transformations (Cohen & Welling, 2016; Weiler & Cesa, 2019). Among these symmetries, rotation is particularly important in scientific and medical imaging. In many tasks, the orientation of an object or tissue structure is not semantically meaningful, but rather a nuisance factor induced by sample preparation, acquisition geometry, or imaging conditions.

This issue appears across several imaging modalities. In digital pathology, tissue micro-structures are scanned at arbitrary orientations, and rotation-equivariant CNNs have been shown to improve patch-level cancer detection and response stability in histopathology (Veeling et al., 2018; Winkens et al., 2018). In microscopy and cell imaging, cellular morphologies and subcellular structures also appear at random orientations, where rotation-equivariant and rotation-invariant architectures have shown consistent gains in image analysis tasks (Chidester et al., 2019; Lafarge et al., 2021). In such settings, a model should ideally respond consistently to the same biological or anatomical structure regardless of its in-plane orientation. Equivariant architectures provide a natural way to enforce this behavior.

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.

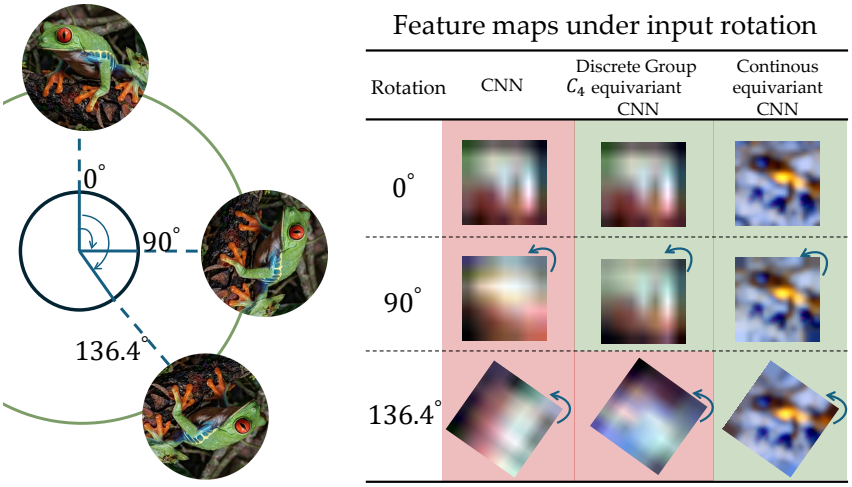


Figure 1: **Rotation consistency of equivariant maps.** *Left:* an input frog image is shown under rotations. *Right:* we compare the feature maps produced by a standard CNN, a discrete group-equivariant CNN (G-CNN) under C_4 (90° rotations), and our rotation-equivariant method CRE. To assess equivariance, we inverse-rotate each feature map back to a common reference frame; aligned responses indicate equivariant behavior. Green backgrounds highlight consistent feature maps, while red backgrounds highlight misaligned responses. The standard CNN produces inconsistent responses under rotation. The G-CNN is equivariant only to rotations contained in its discrete group and therefore fails at intermediate angles, such as 136.4°. In contrast, CRE preserves coherent feature maps across arbitrary rotations.

Existing approaches have made substantial progress toward rotation-aware image models, but they often make one of three compromises: they discretize the rotation group, constrain filters through predefined analytic bases, or move the computation away from the native image grid. Group-equivariant CNNs (Cohen & Welling, 2016), share features over a finite set of rotations, but their equivariance is limited to sampled group elements. Steerable and harmonic methods handle arbitrary rotations through analytic filter bases (Worrall et al., 2017; Weiler & Cesa, 2019), but their expressivity depends on basis design, representation choices, and truncation. Other methods transform the input, lift features to roto-translation spaces, or use learned reference frames and attention-based group operations (Tai et al., 2019; Bekkers et al., 2018; Du et al., 2025; Xu et al., 2023; Islam et al., 2025; Finzi et al., 2020). Although effective, these approaches can add computation, rely on sampled orientations, or depart from the standard image-grid CNN pipeline. This is attractive because native image-grid features preserve direct compatibility with standard CNN pipelines (He et al., 2016; Ronneberger et al., 2015) and dense pixel-aligned prediction, without introducing an explicit orientation dimension.

These limitations motivate CRE, a basis-free local image-grid operator that avoids predefined rotation groups, steerable bases, and explicit orientation lifting. CRE replaces fixed Cartesian kernel weights with center-relative interactions and invariant neighborhood attention. By using shared, learned functions and rotation-invariant distances, it promotes coherent feature responses across arbitrary rotations, including angles outside discrete rotation groups, as illustrated in Figure 1.

Our method is designed as an alternative to standard local convolutional processing, popularly used in natural sciences. It preserves the practical advantages of CNNs, while improving rotational consistency. Empirically, we evaluate the proposed approach on classification and segmentation benchmarks spanning natural images, microscopy, and medical imaging. Our main contributions are:

- We propose CRE, a basis-free image-grid operator using shared center-neighbor interactions and invariant neighborhood attention, without steerable bases or orientation lifting.
- We evaluate rotational consistency across arbitrary test-time rotation angles, showing that discrete and standard baselines often exhibit pronounced oscillatory accuracy between sampled orientations, while CRE displays minimal accuracy variance across rotations.
- CRE plugs directly into existing CNN pipelines for both classification and segmentation, requiring no task-specific modifications — unlike prior baselines.

2 Related Work

CNNs and canonical vision architectures. Convolutional neural networks (CNNs) are the foundation of modern computer vision architectures for classification, recognition, and dense prediction. Early CNNs exploited locality, weight sharing, and translation equivariance to reduce sample complexity and improve scalability (LeCun et al., 1998; Krizhevsky et al., 2012), nowaday several methods still rely on cnn based architectures (Isensee et al., 2021; Stringer et al., 2021; Archit et al., 2025). These principles remain central in canonical architectures such as ResNet (He et al., 2016), which improves optimization through residual connections, and U-Net (Ronneberger et al., 2015), which is widely used for biomedical segmentation through its encoder–decoder structure. However, these models inherit the same core limitation: their built-in equivariance is limited to translations, leaving rotations to be handled indirectly through data, training, or downstream architectural choices. Our work addresses this missing rotation structure while preserving the local image-grid computation.

Data augmentation. Data augmentation is the standard practical approach for improving robustness to rotations (Shorten & Khoshgoftaar, 2019). However, augmentation learns rotational robustness from examples rather than enforcing it by construction, which can require more data, longer training, and careful coverage of the rotation space. It also provides no guarantee that intermediate feature maps transform predictably under rotations. Our method instead builds the symmetry into the local operator, reducing reliance on explicitly sampled rotated examples.

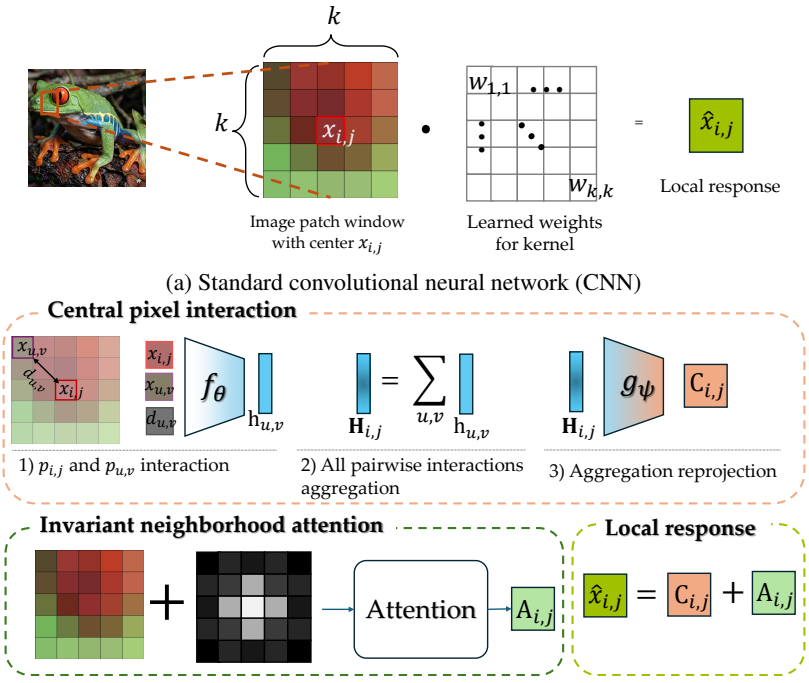
Equivariant models for vision. Equivariant vision models handle rotations through discrete groups, steerable filters, input transformations, lifted spaces, symmetric convolutions, or equivariant attention. First, G-CNN (Cohen & Welling, 2016) extends convolution to finite rotation and reflection groups, but remains exact only for sampled transformations. Later, harmonic and steerable CNNs (Worrall et al., 2017; Cohen & Welling, 2017; Weiler et al., 2018; Weiler & Cesa, 2019) use analytic or steerable filter bases, while SRE (Du et al., 2025) imposes rotation symmetry through symmetric convolutions constrained to basis choices or structured kernel parameterizations. Equivariant transformer (EquiTrans) (Tai et al., 2019) instead transforms the input image, whereas roto-translation covariant CNN (Bekkers et al., 2018)(SE2CNN) and related $\mathrm{SE}(2)$ models lift features to position-orientation spaces (Lafarge et al., 2021). More recent transformer-based methods, such as the equivariant vision transformer (EViT) (Xu et al., 2023) and the Platonic Transformer (Islam et al., 2025), incorporate equivariance through group-aware attention or discrete reference frames. In contrast, our approach targets rotation-consistent local processing directly on the native image grid, without fixed analytic bases, orientation sampling, or explicit lifting.

Continuous-group methods on irregular domains. Continuous-group equivariant methods extend convolutional ideas to general Lie groups and non-Euclidean or irregular domains. LieConv (Finzi et al., 2020), for example, formulates equivariant operations for arbitrary matrix Lie groups and is well suited to sets, point clouds, and graphs with explicit coordinates. These methods are attractive because they can represent continuous symmetries in a principled way, including $\mathrm{SO}(2)$ and $E(2)$ actions. However, they usually require coordinate-based processing, neighborhood search, or irregular-domain operators, making them less direct substitutes for standard dense image convolutions. Our work focuses on continuous rotation-aware local processing within the conventional image-CNN pipeline, avoiding both irregular domain conversion and discretized rotation groups.

Existing methods obtain rotation handling through data augmentation, discrete group sharing, fixed steerable bases, lifted orientation spaces, or coordinate-based continuous group operators. Each strategy introduces a trade-off between equivariance, flexibility, computational cost, and compatibility with standard image-grid CNNs. We propose a basis-free local operator that remains on the native image grid while improving robustness and consistency across arbitrary rotations.

3 Methods

Convolutional neural networks are naturally equivariant to translations because the same kernel is applied at every spatial location. Given an input image or feature map $x \in \mathbb{R}^{C\times H\times W}$, a standard convolution uses a learned kernel of size $k \times k$ and slides it across the image with shared weights. Let (i, j) denote the spatial center of the kernel window, and let (u, v) denote offsets relative to this center. For an odd kernel size k, we write $r = (k - 1)/2$ for the kernel radius, so that $u, v \in \{-r, \ldots, r\}$.



(a) Standard convolutional neural network (CNN)

(b) Continuous rotation equivariant CNN

Figure 2: **Standard 2D convolution and our local invariant 2D convolution. (a)** A $k \times k$ window computes the local response at location (i, j) by applying a position-dependent kernel. Each neighbor is multiplied by a fixed weight; as a result, the operation depends on the grid position of each neighbor. **(b)** Our local invariant operator replaces fixed offset-dependent weights with learned interactions based on relative geometry, and produces the output feature $\hat{x}_{i,j}$ by combining two complementary branches: **Central pixel interaction:** computes a pairwise interaction for the center feature $\mathbf{x}_{i,j}$ and each neighbor $\mathbf{x}_{u,v}$, using the relative distance encoding $d_{u,v}$. This yields hidden interaction features $\mathbf{h}_{u,v}$, which are aggregated into a local representation $\mathbf{H}_{i,j}$ and then reprojected to produce the central interaction $\mathbf{C}_{i,j}$. **Invariant neighborhood attention** ($\mathbf{A}_{i,j}$): the local features within the same window are added with their relative distance and processed by a self-attention module.

The local convolutional response at location (i, j) is given by

$$\hat{x}_{i,j} = \sum_{u=-r}^{r} \sum_{v=-r}^{r} w_{u,v}\, x_{i+u,\, j+v}, \tag{1}$$

where $w_{u,v}$ is the learnable weight associated with the position (u, v) inside a kernel window, standard convolutions preserve translation equivariance since the weights $w_{u,v}$ are reused at every location.

We aim to construct a convolutional block that keeps the desirable translation equivariance of standard CNNs while making the local response stable under arbitrary rotations. We do so without relying on rotation-based data augmentation or predefined steerable filter bases. Our method is illustrated in Figure 2b.

We represent a planar rotation by an angle θ as $R_\theta \in \mathrm{SO}(2)$, acting on spatial coordinates. For a transformation g, equivariance requires that transforming the input and then applying the feature map gives the same result as applying the feature map and then transforming the output representation. In particular, for translation and rotation we require

$$f(L_t x) = L_t f(x), \qquad f(L_R x) = \Pi_R f(x), \tag{2}$$

where $L_t x(p) = x(p - t)$, $L_R x(p) = x(R^{-1} p)$, $p = (i, j)$ is a spatial grid location and Π_R denotes the induced action of the rotation R on the output feature space. In the scalar case, $\Pi_R = L_R$.

A standard sufficient condition for rotation equivariance is that the convolutional kernel itself is invariant under rotations. We provide the proof in Appendix A.1. Based on this principle, we introduce CRE a convolutional block composed of two parts: *central pixel interaction* and *invariant neighborhood attention*.

3.1 CENTRAL PIXEL INTERACTION

The first component models the relation between the center of a local window and its surrounding neighborhood. Let (i, j) denote the center location of a $k \times k$ kernel window, and let (u, v) denote offsets relative to this center, with $u, v \in \{-r, \ldots, r\}$ and $r = (k-1)/2$. The neighboring location associated with offset (u, v) is therefore $(i + u, j + v)$.

Instead of assigning each neighbor a fixed weight based only on its Cartesian offset, as in a standard convolution, we compute a shared interaction between the central feature and each neighboring feature. This interaction is shared across all offsets and spatial locations, preserving translation equivariance while reducing dependence on absolute orientation inside the kernel window.

Let $\mathbf{x}_{i,j} \in \mathbb{R}^{d_{\text{in}}}$ denote the center feature and $\mathbf{x}_{i+u,j+v} \in \mathbb{R}^{d_{\text{in}}}$ denote the neighboring feature at relative offset (u, v). We encode their relative distance as

$$d_{u,v} = \frac{\sqrt{u^2 + v^2}}{\sqrt{2}\, r}, \qquad \phi_d(d_{u,v}) = W_d d_{u,v} + b_d, \tag{3}$$

where $d_{u,v} \in [0, 1]$, with $d_{u,v} = 0$ at the center and $d_{u,v} = 1$ at the kernel corner. The projection ϕ_d maps the scalar distance to a learnable embedding. where $W_d \in \mathbb{R}^{n_d \times 1}$ and $b_d \in \mathbb{R}^{n_d}$. This produces an n_d-dimensional embedding that allows the model to learn how radial distance should affect the local interaction. We use the Euclidean distance from the center because it is invariant under in-plane rotations: rotating the offset (u, v) changes its Cartesian coordinates but preserves its norm. This gives the model relative geometric information about neighborhood.

For each offset (u, v), we compute a pairwise interaction between the center feature and the neighboring feature using a shared function f_θ:

$$\mathbf{h}_{i,j}^{u,v} = f_\theta\left([\mathbf{x}_{i,j} \,\|\, \mathbf{x}_{i+u,j+v} \,\|\, \phi_d(d_{u,v})]\right), \tag{4}$$

where $[\cdot\|\cdot]$ denotes concatenation. The function $f_\theta : \mathbb{R}^{2d_{\text{in}}+n_d} \to \mathbb{R}^D$ maps the concatenated center-neighbor representation to a hidden interaction vector. The parameters θ are shared across all offsets (u, v) and all center locations (i, j), preserving translation equivariance.

The pairwise module produces one hidden vector for each position in the local window. We aggregate these vectors with a permutation-invariant mean and project the result to the output dimension:

$$\mathbf{H}_{i,j} = \frac{1}{k^2} \sum_{u=-r}^{r} \sum_{v=-r}^{r} \mathbf{h}_{i,j}^{u,v}, \qquad \mathbf{C}_{i,j} = g_\psi(\mathbf{H}_{i,j}). \tag{5}$$

Here, $\mathbf{H}_{i,j} \in \mathbb{R}^D$ is the aggregated local representation, $g_\psi : \mathbb{R}^D \to \mathbb{R}^{d_{\text{out}}}$ is a shared projection function, and $\mathbf{C}_{i,j} \in \mathbb{R}^{d_{\text{out}}}$ is the central pixel interaction feature.Since both branches share parameters across locations and depend only on center-relative features and rotation-invariant distances, the operator is translation equivariant and rotation-consistent by construction.

Implementation. In practice, both f_θ and g_ψ are implemented as small multilayer perceptrons. The interaction function f_θ consists of a linear layer followed by a LeakyReLU activation:

$$f_\theta = \text{LeakyReLU} \circ \text{Linear}\left(2d_{\text{in}} + n_d,\, 2d_{\text{out}}\right). \tag{6}$$

The projection function g_ψ maps the aggregated interaction feature to the output channel dimension using a two-layer MLP with a negative slope of 0.1 for the LeakyReLU activation:

$$g_\psi = \text{Linear}\left(d_{\text{out}},\, d_{\text{out}}\right) \circ \text{LeakyReLU} \circ \text{Linear}\left(2d_{\text{out}},\, d_{\text{out}}\right). \tag{7}$$

Our aggregation follows the principle of permutation-invariant set processing (Zaheer et al., 2017; Qi et al., 2017) and message passing Gilmer et al. (2017). Unlike these works, we keep the computation on dense image-grid neighborhoods and use center-relative distance encodings to target rotation-consistent local image features. In contrast to filters derived from fixed basis functions, our use of MLPs enables CRE to learn arbitrary kernel functions.

3.2 Invariant Neighborhood Attention

The central pixel interaction module models the relation between the center pixel and each neighbor independently. However, this does not capture interactions among different neighbors within the same local window. To model these higher-order local relationships, we introduce an invariant neighborhood attention module, where all pixels in the kernel window attend to one another.

Let $\mathcal{N}_{i,j} = \{\mathbf{x}_{i+u,j+v} : u, v \in \{-r, \dots, r\}\}$ denote the local neighborhood around (i, j). Before applying attention, we add a distance-based positional embedding to each local feature. Specifically, for each offset (u, v), we use the normalized radial distance $d_{u,v}$ from Eq. 3 and compute

$$\tilde{\mathbf{x}}_{i+u,j+v} = \mathbf{x}_{i+u,j+v} + \phi_p(d_{u,v}), \tag{8}$$

where $\phi_p : \mathbb{R} \to \mathbb{R}^{d_{\text{in}}}$ is a learnable MLP. This encoding provides relative geometric information, avoiding absolute Cartesian position. We then prepend a learnable class token $\mathbf{x}_{\text{cls}} \in \mathbb{R}^{d_{\text{in}}}$ and define

$$\mathcal{T}_{i,j} = \{\mathbf{x}_{\text{cls}}\} \cup \{\tilde{\mathbf{x}}_{i+u,j+v} : u, v \in \{-r, \dots, r\}\}.$$

Each token in $\mathcal{T}_{i,j}$ is projected to query, key, and value representations. Stacking them gives $Q, K, V \in \mathbb{R}^{(k^2+1)\times d}$, where d is the attention dimension. We compute local self-attention, extract the updated class token, and project it to the output dimension:

$$O_{i,j} = \text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \qquad \mathbf{h}_{\text{cls}} = O^{(0)}_{i,j}, \qquad \mathbf{A}_{i,j} = W_o\mathbf{h}_{\text{cls}} + b_o. \tag{9}$$

Here, $O^{(0)}_{i,j} \in \mathbb{R}^d$ is the first row of $O_{i,j}$, corresponding to the class token, and $\mathbf{A}_{i,j} \in \mathbb{R}^{d_{\text{out}}}$ is the invariant neighborhood attention feature at location (i, j). Because the positional embedding depends only on the radial distance to the center, the attention module retains relative geometric information. Since the class token is permutation invariant it will also be rotation equivariant.

Unlike patch-level attention methods such as EViT (Xu et al., 2023)and NAT (Hassani et al., 2023), our module operates on pixels. It uses normalized Euclidean distance rather than learned or offset-based positional encodings, and a `[CLS]` token to aggregate information.

Local response The final output at center location (i, j) is obtained by combining the central pixel interaction feature $\mathbf{C}_{i,j}$ (Eq. 5) with the invariant neighborhood attention feature $\mathbf{A}_{i,j}$ (Eq. 9). Both terms lie in $\mathbb{R}^{d_{\text{out}}}$. We define the local response as $\hat{x}_{i,j} = \mathbf{C}_{i,j} + \mathbf{A}_{i,j}$. Since both branches share parameters across spatial locations and depend only on center-relative features and rotation-invariant distances, their outputs keeps equivariance under translations and rotations.

4 Experiments

All models are trained without rotation augmentation. Detailed information about datasets and usage is in Appendix A.3. Implemented network architectures are reported in Appendix A.4, and optimization details, stopping criteria, and loss functions are described in Appendix A.2.

4.1 Classification

We compare our method with standard and rotation-equivariant baselines on three benchmarks, evaluating both accuracy and the stability of performance under continuous input rotations.

Experimental Setup

Baselines. We use Standard CNN (LeCun et al., 1998) as the non-equivariant baseline. Rotation-aware baselines cover the main design families: discrete group convolutions (G-CNN (Cohen & Welling, 2016)), harmonic and steerable filters (Harmonic (Worrall et al., 2017); Steerable (Weiler & Cesa, 2019) CNN), input transformations (EquiTrans (Tai et al., 2019)), roto-translation modeling (SE2CNN (Bekkers et al., 2018)), symmetric convolutions (SRE (Du et al., 2025)), and reference-frame lifting with attention (Platonic Transformer (Islam et al., 2025)).

Datasets and rotation protocol. We evaluate on **MNIST** (Larochelle et al., 2007), **CIFAR-10** (Krizhevsky et al., 2009), and **PathMNIST** (Yang et al., 2023), covering shape recognition, natural-image classification, and orientation-variable medical imaging, respectively. Test images are rotated by $\theta \in 0^\circ, 5^\circ, \dots, 355^\circ$ using bilinear interpolation, with boundaries filled by the per-channel image mean to avoid constant-border artifacts.

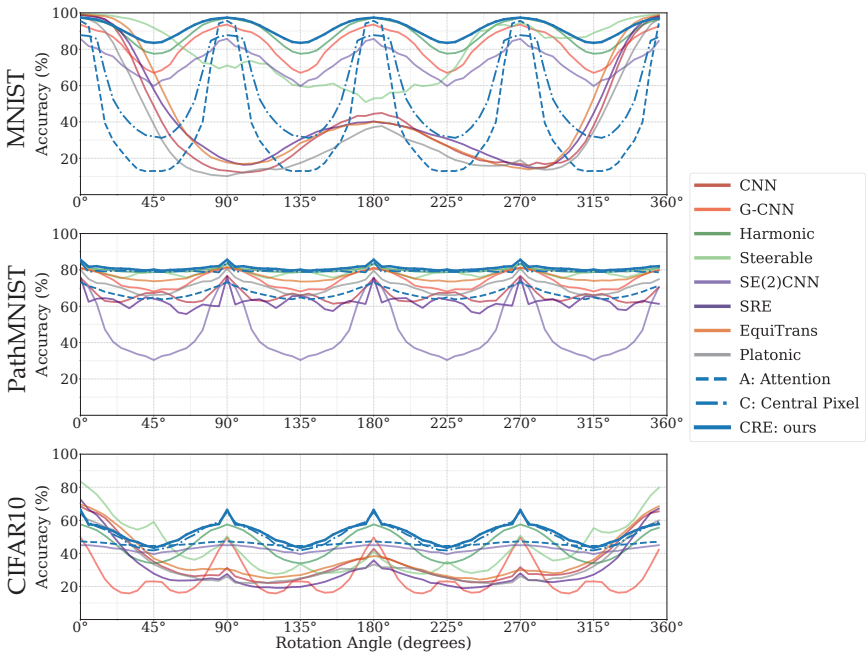


Figure 3: **Classification accuracy under continuous input rotations.** We report classification accuracy as a function of test-time rotation angle on MNIST, PathMNIST, and CIFAR-10. Blue curves denote our CRE model and its ablated components while other colors denote baseline methods. A flatter curve indicates better rotational consistency. Across the three datasets, our method provides the most flat response. On PathMNIST, it achieves the best accuracy. Overall, CRE offers the best trade-off between mean accuracy and rotational equivariance.

Rotational Consistency Results

Accuracy across continuous rotation angles. Rather than focusing on peak accuracy, we analyze the accuracy curve for rotation angles from $0°$ to $360°$ (Figure 3). A more rotation-equivariant model should produce a flatter curve, indicating similar performance across angles .We quantify rotational stability using the variance (Var) of accuracy across test angles, $\sigma_\theta^2 = \mathrm{Var}_\theta[\mathrm{Acc}(\theta)]$, where lower values indicate more consistent performance under rotation.

Discrete equivariant methods, such as G-CNN (Cohen & Welling, 2016) with C_8 symmetry, show more oscillatory curves, with peaks near sampled orientations and drops in between. This suggests that their equivariance is strong at discretized angles. In contrast, CRE achieves the highest mean accuracy on CIFAR-10, MNIST, and PathMNIST, while maintaining low rotational variance (Table 1). On CIFAR-10, CRE improves accuracy over the baselines across most rotated inputs. On PathMNIST, CRE also achieves the best performance at the canonical $0°$ orientation, indicating that the proposed operator does not sacrifice base-orientation accuracy in this domain. Notably, the some baselines also exhibits angle-dependent variation on MNIST and CIFAR-10, showing that analytic continuous-equivariance constructions do not necessarily yield uniform discrete-grid performance.

At $\theta = 0°$, some baselines may still perform better because they can specialize to the upright training distribution. However, this specialization often leads to larger drops at non-canonical rotations. CRE instead provides a stronger balance between accuracy and stability, yielding the best mean accuracy over the full rotation range and competitive or superior rotational equivariance across datasets.

Ablation Study. We ablate the two core components of our CRE to measure their effects on accuracy and rotational consistency. We compare three variants: the central pixel interaction ($\mathbf{C}$), the invariant attention ($\mathbf{A}$), and the full model (**CRE**:$\mathbf{C} + \mathbf{A}$). These variants are shown in blue in Figure 3. Both branches improve rotational consistency, and their combination achieves the better accuracy curve. Removing $\mathbf{C}$ causes a larger performance drop than removing $\mathbf{A}$, nevertheless, $\mathbf{A}$ seems to be more equivariant with lower σ^2. Notably, both single-branch results indicate that the observed equivariance comes from the principled local invariant design.

Table 1: **Classification performance aggregated over rotation angles.** We report mean accuracy and rotational variance (Var) across angles for MNIST, PathMNIST, and, CIFAR-10. Higher mean accuracy (**bold**) indicates better overall classification performance, while lower variance (red) indicates stronger rotational consistency. CRE achieves the best mean accuracy on all three datasets and the lowest variance on MNIST, while remaining competitive in variance on CIFAR-10 and PathMNIST. CRE improves average performance while maintaining stable behavior under continuous rotations.

Method	MNIST			pathmnist			CIFAR10		
	Mean Acc.	Var	Best Acc.	Mean Acc.	Var	Best Acc.	Mean Acc.	Var	Best Acc.
Standard CNN (LeCun et al., 1998)	39.55	733.88	98.97	66.00	14.02	76.81	34.46	161.97	68.43
G-CNN (Cohen & Welling, 2016)	82.29	79.46	93.54	73.98	19.84	81.21	25.32	101.14	49.60
Harmonic (Worrall et al., 2017)	88.38	49.27	97.38	79.78	**1.38**	83.38	44.95	66.30	57.54
Steerable (Weiler & Cesa, 2019)	78.76	243.94	**99.48**	78.57	4.32	83.06	42.85	214.83	**83.46**
SE2CNN (Bekkers et al., 2018)	42.57	694.87	98.85	62.53	16.06	76.27	30.00	175.48	72.62
SRE (Du et al., 2025)	72.76	59.37	85.69	45.54	211.31	73.93	42.72	2.82	44.96
Equivariant transformer (Tai et al., 2019)	43.40	817.37	99.22	76.61	5.91	81.60	36.39	159.63	69.54
Platonic (Islam et al., 2025)	34.04	675.63	97.86	70.93	14.70	80.67	32.12	133.04	63.39
C: Central Pixel	55.58	475.54	87.75	79.45	1.54	84.47	50.64	45.14	64.90
A: Attention	39.46	873.96	95.61	67.02	7.94	73.12	45.71	**0.89**	47.32
CRE: ours	**91.26**	**23.92**	97.38	**81.00**	1.83	**85.60**	**52.11**	34.16	66.30

Appendix 6 provides a detailed efficiency comparison, showing CRE achieves its improved rotational consistency with moderate parameter count but at increased memory and runtime cost, although our high-level implementation can be further optimized for memory and latency (Sec. 5).

4.2 Segmentation

We evaluate CRE on dense prediction tasks, where rotation consistency must hold at the pixel level and preserve spatial structure and boundaries.

Experimental Setup

Baselines. We compare against a standard U-Net (Ronneberger et al., 2015) baseline for semantic segmentation. We also include Steerable (Weiler & Cesa, 2019) and SE2CNN (Bekkers et al., 2018), which are among the few equivariant architectures which support dense segmentation.

Datasets. We use three segmentation benchmarks from microscopy and medical imaging. The **ISBI** 2012 EM segmentation challenge (Arganda-Carreras et al., 2015; Bekkers et al., 2018; Weiler et al., 2018) contains electron microscopy images with dense neuronal membrane annotations. The **DRIVE Retina** dataset (Staal et al., 2004; Bekkers et al., 2018) contains color fundus images with pixel-wise blood vessel annotations, where vessels appear at arbitrary orientations and scales. The **Fluorescent neuronal cells** dataset (Clissa et al., 2021) contains fluorescent microscopy images of labeled neurons with annotations for cell segmentation, making it a natural testbed for rotation-robust segmentation. For segmentation, we do not use a rotation protocol: both train and test splits already contain structures at mixed orientations, matching the standard evaluation used by prior baselines. We train on image patches, which increases the effective sample size and is convenient for small datasets.

Evaluation Metrics. We evaluate segmentation using binary cross entropy (BCE) loss and the receiver operating characteristic (ROC) curve, following prior work (Bekkers et al., 2018; Weiler et al., 2018; Arganda-Carreras et al., 2015). For ISBI 2012, we additionally report the adapted Rand score (V_{rand}) and the variation-of-information score (V_{info}), which are standard connectomics metrics (Weiler et al., 2018; Arganda-Carreras et al., 2015). We do not report these metrics for retina and neuronal cells because they are not scoped as connected-components.

Segmentation performance.

Figure 4 and Table 2 show that CRE is competitive with the equivariant baselines, while rotation-aware models generally improve over the standard U-Net (Ronneberger et al., 2015). On ISBI, CRE better captures small membrane holes and enclosed structures, but can introduce spurious openings in ambiguous regions. On retina, CRE best preserves the vessel tree structure; its vessels are thicker than those of Steerable (Weiler & Cesa, 2019) and SE2CNN (Bekkers et al., 2018), but these baselines miss more thin branches. On fluorescent neuronal cells, Steerable (Weiler & Cesa, 2019) better suppresses axon-like structures but often fails to close soma boundaries. CRE and SE2CNN (Bekkers et al.,

Table 2: Segmentation results across datasets. We report BCE loss (↓), ROC (↑), V_{rand} (↑), and V_{info} (↑). Best results are shown in **bold**.

	ISBI 2012				DRIVE Retina		Fluorescent Neuronal Cells	
Model	BCE ↓	ROC ↑	V_{rand} ↑	V_{info} ↑	BCE ↓	ROC ↑	BCE ↓	ROC ↑
U-Net (Ronneberger et al., 2015)	0.3053	0.9484	0.9048	0.6827	0.2284	0.8825	0.0218	0.9842
Steerable (Weiler & Cesa, 2019)	**0.2603**	**0.9600**	**0.9297**	**0.7319**	0.3520	0.7931	**0.0124**	**0.9944**
SE2CNN (Bekkers et al., 2018)	0.3271	0.9350	0.7891	0.5036	0.4238	0.8443	0.0150	0.9911
CRE (ours)	0.3265	0.9363	0.8697	0.5824	**0.1453**	**0.9541**	0.0168	0.9936

2018) behave similarly: CRE tends to under-segment, whereas SE2CNN (Bekkers et al., 2018) more often over-segments or merges nearby cells.

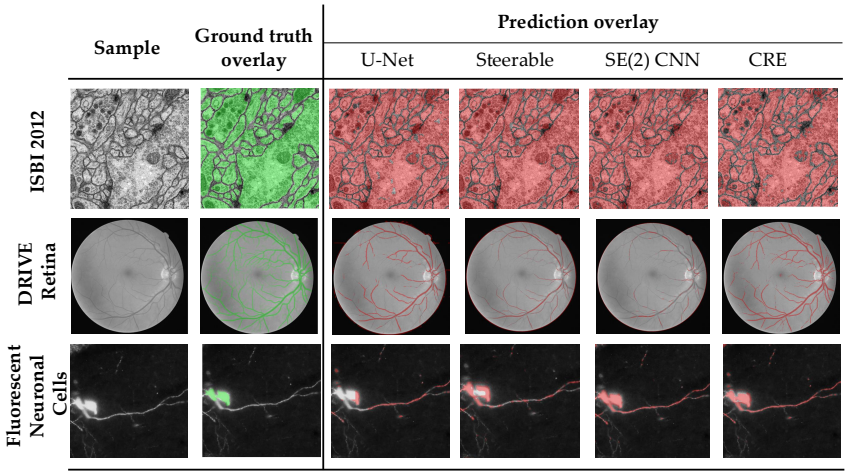


Figure 4: **Qualitative segmentation results.** We show one randomly selected test sample from each dataset. For each example, we report the original input image, the ground-truth mask overlay (green), and the model predictions overlaid (red). The models produce broadly similar segmentations, but differ in fine details such as thin structure recovery, spurious openings, and cell merging.

5 Conclusion

We introduced CRE, a basis-free local operator for rotation-consistent processing directly on the native image grid. In classification, CRE achieves the best mean accuracy across datasets under continuous test-time rotations, while maintaining competitive or superior rotational variance (Figure 3, Table 1). The ablation shows that both components contribute: center-pixel interactions provide geometry-aware aggregation, while attention adds complementary neighborhood context. Together, they achieve strong mean accuracy with low angle-dependent variation.

We also evaluated CRE on segmentation, where dense predictions require pixel-level spatial consistency. CRE is competitive with established equivariant baselines and performs particularly well on the retina dataset, where it best preserves vessel-tree structure (Figure 4, Table 2). These results suggest that CRE can be used beyond classification contrasting with many equivariant baselines primarily evaluated on classification.

CRE may benefit image-based learning systems by improving robustness to rotation. However, it should not be deployed in high-stakes applications without task-specific validation, since untested distribution shifts could affect downstream decisions.

Limitations and future work. Our experiments focus on in-plane rotations and two common vision tasks. The current implementation of CRE is not yet optimized for low-level GPU execution. Standard convolutions benefit from highly optimized CUDA kernels and years of engineering, whereas our implementation relies on less optimized local-window operations and attention computations. Therefore, the measured overhead likely reflects both the operator itself and the current implementation. Future work will investigate fused kernels, improved memory reuse, and more efficient local attention to reduce this gap. Future work will focus on improving efficiency and broader evaluation on additional tasks such as detection, registration, and high-resolution segmentation.

AI DISCLOSURE

The authors used generative AI tools for limited edition support, including grammar correction, readability improvements, and basic LaTeX assistance. AI tools were also used to help diagnose issues in code written by the authors.

They were not used to produce research code, generate references, develop the methodology, analyze results, or draft full sections of the manuscript. All AI-assisted material was checked by the authors, who remain responsible for the final content.

REFERENCES

Anwai Archit, Luca Freckmann, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Carolin Teuber, Melanie Spitzner, Constanza Tapia Contreras, et al. Segment anything for microscopy. *Nature methods*, 22(3):579–591, 2025.

Ignacio Arganda-Carreras, Srinivas C. Turaga, Daniel R. Berger, Dan Cireşan, Alessandro Giusti, Luca M. Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Shashi N. Dwivedi, Joachim M. Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9:142, 2015. doi: 10.3389/fnana.2015.00142.

Erik J. Bekkers, Maxime W. Lafarge, Mitko Veta, Koen A. J. Eppenhof, Josien P. W. Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2018*, volume 11070 of *Lecture Notes in Computer Science*, pp. 440–448. Springer, 2018.

Benjamin Chidester, Tianming Zhou, Minh N. Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 2019.

Luca Clissa, Roberto Morelli, Fabio Squarcio, Timna Hitrec, Marco Luppi, Lorenzo Rinaldi, Matteo Cerri, Roberto Amici, Stefano Bastianini, Chiara Berteotti, Viviana Lo Martire, Davide Martelli, Alessandra Occhinegro, Domenico Tupone, Giovanna Zoccoli, and Antonio Zoccoli. Fluorescent neuronal cells. Unpublished, 2021. URL `https://amsacta.unibo.it/id/eprint/6706/`.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, NY, USA, 2016. PMLR.

Taco S Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations (ICLR)*, 2017.

Yuexi Du, Jiazhen Zhang, Tal Zeevi, Nicha C. Dvornek, and John A. Onofrey. SRE-conv: Symmetric rotation equivariant convolution for biomedical image classification. *arXiv preprint arXiv:2501.09753*, 2025. URL `https://arxiv.org/abs/2501.09753`.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3136–3146. PMLR, 2020.

Castañeda Alejandro Garcia, Jan van Gemert, Daan Brinks, and Nergis Tömen. Learning physics from video: Unsupervised physical parameter estimation for continuous dynamical systems. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6185–6194, June 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. IEEE, 2016.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Mohammad Mohaiminul Islam, Rishabh Anand, David R Wessels, Friso de Kruiff, Thijs P Kuipers, Rex Ying, Clara I Sánchez, Sharvaree Vadgama, Georg Bökman, and Erik J Bekkers. Platonic transformers: A solid choice for equivariance. *arXiv preprint arXiv:2510.03511*, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

Maxime W Lafarge, Erik J Bekkers, Josien PW Pluim, Remco Duits, and Mitko Veta. Roto-translation equivariant convolutional networks: Application to histopathology image analysis. *Medical Image Analysis*, 68:101849, 2021.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Luke Oakden-Rayner. Exploring large-scale public medical image datasets. *Academic radiology*, 27 (1):106–112, 2020.

Esther Puyol-Antón, Bram Ruijsink, Stefan K Piechnik, Stefan Neubauer, Steffen E Petersen, Reza Razavi, and Andrew P King. Fairness in cardiac mr image analysis: an investigation of bias due to data imbalance in deep learning based segmentation. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 413–423. Springer, 2021.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net:convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015. URL https://arxiv.org/abs/1505.04597.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Joes Staal, Michael D. Abràmoff, Meindert Niemeijer, Max A. Viergever, and Bram van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004. doi: 10.1109/TMI.2004.825627.

Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant transformer networks. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1901.11399.

Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016. doi: 10.1109/TMI.2016.2535302.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. *arXiv preprint arXiv:1806.03962*, 2018.

Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL `https://papers.neurips.cc/paper/by-source-2019-8112`.

Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 849–858, 2018.

Jim Winkens, Jasper Linmans, Bastiaan S. Veeling, Taco Cohen, and Max Welling. Improved semantic segmentation for histopathology using rotation equivariant networks. In *ICLR 2018 Workshop Track*, 2018.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5028–5037. IEEE, 2017.

Renjun Xu, Kaifan Yang, Ke Liu, and Fengxiang He. $e(2)$-equivariant vision transformer. In *Uncertainty in artificial intelligence*, pp. 2356–2366. PMLR, 2023.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific data*, 10(1):41, 2023.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 PROOFS

Proposition 1 (Rotationally invariant kernels yield rotation-equivariant convolutions)**.** *Let $x : \mathbb{R}^d \to \mathbb{R}$, and define the convolution operator*

$$(Fx)(u) = \int_{\mathbb{R}^d} K(r)x(u-r)\,dr, \tag{10}$$

where $K : \mathbb{R}^d \to \mathbb{R}$. Let $R \in SO(d)$ act on signals by

$$(L_R x)(u) = x(R^{-1}u). \tag{11}$$

If the kernel is rotationally invariant, namely

$$K(Rr) = K(r) \qquad \forall r \in \mathbb{R}^d,\ \forall R \in SO(d), \tag{12}$$

then F is rotation equivariant:

$$F(L_R x) = L_R F(x). \tag{13}$$

Proof. For any $u \in \mathbb{R}^d$,

$$F(L_R x)(u) = \int_{\mathbb{R}^d} K(r)(L_R x)(u-r)\,dr \tag{14}$$

$$= \int_{\mathbb{R}^d} K(r)x(R^{-1}(u-r))\,dr \tag{15}$$

$$= \int_{\mathbb{R}^d} K(r)x(R^{-1}u - R^{-1}r)\,dr. \tag{16}$$

Using the change of variables $s = R^{-1}r$, so that $r = Rs$, and since $R \in SO(d)$ has determinant one, $dr = ds$. Therefore,

$$F(L_R x)(u) = \int_{\mathbb{R}^d} K(Rs)x(R^{-1}u - s)\,ds \tag{17}$$

$$= \int_{\mathbb{R}^d} K(s)x(R^{-1}u - s)\,ds \tag{18}$$

$$= F(x)(R^{-1}u) \tag{19}$$

$$= (L_R F(x))(u). \tag{20}$$

Thus $F(L_R x) = L_R F(x)$. □

A.2 TRAINING DETAILS

All models were trained on NVIDIA A40 GPUs until convergence. Unless otherwise specified by the original implementation, we used the Adam optimizer with learning rate 10^{-3} and weight decay 10^{-3}. Batch size was selected separately for each dataset–model combination as the largest value that fit within GPU memory, allowing each model to use the available hardware efficiently.

For classification, models were trained with cross-entropy loss. For segmentation, we used a combined binary cross-entropy and Dice loss, defined as $\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}}$. Rotations were used only at test time to evaluate rotational consistency; no rotation augmentation was used during training.

For classification, most CNN-based baselines converged within approximately 100 epochs. CRE and the Platonic Transformer required longer training, typically around 300 epochs, while EViT required more than 600 epochs in our setting. For EViT and the Platonic Transformer, we followed the training requirements and implementation details provided in the original papers and official codebases, since these models require architecture-specific training protocols. For segmentation, all models were trained for 300 epochs, as all methods converged by this point or earlier.

We monitored the training loss and retained the best model according to the lowest observed loss. An epoch was considered an improvement only when $\mathcal{L}_{\text{epoch}} < \mathcal{L}_{\text{best}} - 10^{-4}$. Training was stopped after convergence according to this criterion.

A.3 Dataset Details

Table 3 summarizes the datasets used in our classification and segmentation experiments. For all datasets, we used an 80–20 train–test split. Models were trained on the training split, and the checkpoint with the lowest training loss was selected for evaluation. Rotations were applied only at test time.

Table 3: **Dataset details.** We report the task, imaging domain, input shape, number of samples, and split used in our experiments. Shapes correspond to the standard dataset resolution unless otherwise stated. All datasets use an 80–20 train–test split, and the best checkpoint is selected according to the lowest training loss.

Dataset	Task	Domain	Shape	Samples	Classes	Patch size
MNIST	Classification	Handwritten digits	$1 \times 28 \times 28$	12,000	10	–
CIFAR-10	Classification	Natural images	$3 \times 32 \times 32$	60,000	10	–
PathMNIST	Classification	Histopathology	$3 \times 28 \times 28$	89,996	9	–
ISBI 2012	Segmentation	Electron microscopy	$1 \times 512 \times 512$	60	2	64×64
DRIVE retina	Segmentation	Retinal fundus imaging	$3 \times 584 \times 565$	40	2	64×64
Fluorescent neuronal cells	Segmentation	Fluorescence microscopy	$3 \times 1600 \times 1200$	283	2	64×64

A.4 Network Architectures

We provide the network architectures used in our experiments for completeness and reproducibility. Table 4 describes the classification model, which stacks CRE convolutional blocks with normalization, nonlinearity, intermediate pooling, and a final global classifier. Table 5 describes the segmentation model, which follows a fully convolutional design and preserves spatial resolution for dense pixel-wise prediction. In both cases, CREConv denotes our proposed local convolutional block.

Table 4: **Classification network architecture.** The classifier is built from repeated CRE convolutional blocks with layer normalization and ReLU activations. Max pooling is applied after every two hidden blocks, followed by a final CRE block, global average pooling, and a two-layer linear classifier.

Stage	Operation	Output channels
Input	Image $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$	C_{in}
Initial block	CREConv($k = 5$) + LayerNorm + ReLU	32
Hidden block 1	CREConv($k = 5$) + LayerNorm + ReLU	32
Hidden block 2	CREConv($k = 5$) + LayerNorm + ReLU + MaxPool(2)	64
Hidden block 3	CREConv($k = 5$) + LayerNorm + ReLU	64
Hidden block 4	CREConv($k = 5$) + LayerNorm + ReLU + MaxPool(2)	128
Hidden block 5	CREConv($k = 5$) + LayerNorm + ReLU	128
Final block	CREConv($k = 5$)	256
Pooling	Adaptive average pooling	256
Classifier	Linear$(256, 128)$ + ReLU + Linear$(128, N_{\text{cls}})$	N_{cls}

Table 5: **Segmentation network architecture.** The segmentation model is a fully convolutional network built from repeated CRE convolutional blocks. Each hidden CRE block uses kernel size $k = 5$, hidden dimension 8, layer normalization, and ReLU activation. Unlike the classification model, no spatial pooling is used, so the network preserves the input resolution.

Stage	Operation	Output channels
Input	Image $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$	C_{in}
Initial block	CREConv($k = 5$) + LayerNorm + ReLU	8
Hidden block 1	CREConv($k = 5$) + LayerNorm + ReLU	16
Hidden block 2	CREConv($k = 5$) + LayerNorm + ReLU	16
Hidden block 3	CREConv($k = 5$) + LayerNorm + ReLU	16
Hidden block 4	CREConv($k = 5$) + LayerNorm + ReLU	8
Final block	CREConv($k = 5$)	C_{out}
Output	Dense segmentation map $\hat{y} \in \mathbb{R}^{C_{\text{out}} \times H \times W}$	C_{out}

A.5 EFFICIENCY ANALYSIS.

Table 6 reports the computational cost of each model. All measurements are computed using dummy inputs of shape $8 \times 3 \times 32 \times 32$, simulating a CIFAR-10 batch with batch size 8. The standard CNN is the most efficient baseline, with the smallest inference time and highest throughput, but it lacks rotation handling. Among equivariant baselines, the cost varies substantially: lightweight methods such as SRE (Du et al., 2025)and G-CNN (Cohen & Welling, 2016)remain close to standard CNNs, whereas Harmonic (Worrall et al., 2017), SE2CNN (Bekkers et al., 2018), and especially Platonic (Islam et al., 2025)introduce larger runtime or memory overhead. Platonic (Islam et al., 2025)has by far the largest parameter count and model size, which also reduces its maximum feasible batch size.

CRE has fewer parameters than Steerable (Weiler & Cesa, 2019)and far fewer than Platonic (Islam et al., 2025), but it uses more peak GPU memory and has a slower training step than most compact baselines. This overhead comes from explicitly computing local center-neighbor interactions and attention within each kernel window. The ablations clarify the trade-off: the central pixel branch is faster and uses less memory than the attention branch, while the full CRE model combines both costs. Thus, CRE offers improved rotational consistency and strong accuracy, but its current implementation is not the most computationally efficient. Improving memory reuse, local-window batching, and attention computation is therefore a natural next step.

Table 6: **Computational efficiency of classification models.** We compare parameter count, model size, peak GPU memory, training step time, throughput, inference time, and maximum feasible batch size for all evaluated classification models on the same hardware setting. CRE has a moderate parameter count compared with large equivariant baselines such as Steerable (Weiler & Cesa, 2019)and Platonic (Islam et al., 2025), but requires higher peak memory because it combines central pixel interactions and local attention. The single-branch variants show the source of this cost: the attention branch increases memory, while the central pixel branch adds fewer memory demands but still increases computation. These results show that CRE improves rotation consistency at the cost of additional runtime and memory, leaving efficiency optimization as an important direction for future work.

Model	Params	Size (MB)	Peak GPU Mem. (MB)	Step Time (ms)	Throughput (img/s)	Inference (ms)	Max Batch
Standard CNN (LeCun et al., 1998)	22,490	0.09	258.65	2.376	3,367.3	0.300	8,191
EquiTrans (Tai et al., 2019)	42,891	0.17	269.24	6.403	1,249.5	1.344	8,191
G-CNN (Cohen & Welling, 2016)	24,780	0.12	272.92	6.873	1,164.0	2.804	8,191
Harmonic (Worrall et al., 2017)	34,250	0.17	282.48	50.047	159.8	24.993	8,191
Steerable (Weiler & Cesa, 2019)	2,081,675	8.69	399.09	17.458	458.2	3.667	5,550
SRE (Du et al., 2025)	5,190	0.02	266.31	2.721	2,940.2	0.732	8,191
SE2CNN (Bekkers et al., 2018)	41,166	0.17	270.90	26.023	307.4	11.976	8,191
Platonic (Islam et al., 2025)	19,541,962	74.67	1084.81	64.091	124.8	22.044	872
C: Central Pixel	628,176	2.42	946.36	18.648	429.0	1.670	544
A: Attention	252,988	1.00	1181.76	35.576	224.9	2.278	381
CRE: ours	846,978	3.29	1608.50	52.325	152.9	6.284	257