# Relaxed Equivariance for Anisotropic Noise

**Lucas Steinberger**
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
lsteinberger@fas.harvard.edu

## Abstract

Equivariant neural networks have been shown to be more data-efficient and generalizable than their non-equivariant counterparts when the target function is known to be equivariant to some symmetry group. Recent work has also shown that for some physical systems that have approximate symmetries, approximately equivariant models can outperform both normal and strictly equivariant models, retaining state-of-the-art performance on fully equivariant tasks. In this work, we test the effectiveness of relaxed equivariant CNNs on a rotated MNIST classification task with added isotropic and anisotropic noise. We find that contrary to our hypothesis and previous work, the fully equivariant model outperforms both the relaxed equivariant and normal CNN models in all noise settings.

## 1 Introduction

Neural networks are an incredibly versatile class of models that have seen successful applications to an astonishing variety of tasks. While the theory of how and why neural networks work so successfully is far from sufficient, their broad applicability can be partially understood as a result of the universal approximation theorem [1], which states that a sufficiently wide neural network can approximate any continuous function to arbitrary precision. However, the expressivity of neural networks is both a blessing and a curse, because it means that the search space of possible functions is vast. This makes it difficult to find the right function for a given task, which in turn increases the need for data and compute power.

Taking the view that the goal of training a neural network is to find some unknown target function, it is reasonable to suggest that prior knowledge of the target function should be leveraged to guide or constrain the search. Constructing a strong and useful prior is in general not an easy task, and using human intuition to design priors is time-consuming and risks imposing incorrect assumptions on the model. At the same time, access to large datasets and compute power continues to grow, so for many problems, such as large language models, the dominant approach is to continue scaling up data and model size, rather than to rely on inductive bias [2].

However, there are still many settings, especially in physics and other scientific domains, where there is already strong prior knowledge of the target function due to existing theories, and furthermore data may be scarce and expensive to obtain. Such scenarios call for the development of highly optimized models that are specifically tailored to fit the known properties of the target function. One particularly fruitful avenue of research in this direction has been the development of equivariant neural networks, which became widely popular after the seminal work of Cohen and Welling [3]. While in their original paper, Cohen and Welling do create a new model architecture for 2D images, the underlying idea of equivariance is much more general. An equivariant model is one that respects the symmetries of the target function, meaning that if the input is transformed in some way, the output will transform in a predictable manner, in a way that can be made precise using group theory. For example, if the

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

target function is known to be rotation-equivariant, then rotating the input image should result in a corresponding rotation of the output.

Equivariance had in fact already seen previous success in convolutional neural networks (CNNs), which are equivariant to translations [4], but are often not presented in this way. Cohen and Welling extended these symmetries to rotations and reflections, but since then, equivariant neural networks have been developed for a wide variety of symmetry groups and scientific problems, ranging from 3D point clouds to graph neural networks (GNNs) for interatomic potentials [5, 6].

Equivariant networks have been shown to outperform their non-equivariant counterparts in many settings, and seem to be consistently more data-efficient [3, 5, 6]. This is to be expected, because normal models need to learn the symmetries of the target function from data or data augmentation, while equivariant models have these symmetries built in by design. For a specific example, consider the MNIST dataset, which Cohen and Welling [3] use to demonstrate the benefits of equivariant networks. Consider the task of classifying images of handwritten digits, but where the digits may be rotated by 90 degree increments. In order to learn this task, a normal CNN would need to see many examples of each digit in each possible rotation, while a rotation-equivariant network would only need to see the digits in one orientation, because the model is already constrained to be rotation-equivariant. This results in a significant reduction in the amount of data needed to learn the task.

Of course, equivariance is only beneficial if the target function is indeed equivariant to the chosen symmetry group. An equivariant model is incapable of learning a function that has even a small deviation from the assumed symmetry. There are, however, functions that have a high degree of approximate symmetry but have small yet meaningful deviations from it. For a good example, consider a physical system in a small non-uniform external field, such as a gravitational or electromagnetic field. The underlying physical laws governing the system may be perfectly symmetric, but the presence of the external field breaks the symmetry slightly. This provides motivation for a new class of models that are *approximately* equivariant, in the sense that they are biased towards equivariant functions, hypothetically inheriting some of the data efficiency, but are still capable of learning small asymmetries. This is the focus of a paper from Wang, Walters, and Yu [7], and the focus of this work. In Section 2, we present a quick review of group theory and group equivariant CNNs. In Section 3, we review some of the methods proposed by Wang et al. In Section 4, we outline the experimental question and methodology of this paper with reference to Wang et al. In Section 5 we present our results, and in Section 6 we discuss the implications of our findings and possible future directions.

## 2 Group Theory and Group Equivariant CNNs

A full treatment of group theory is beyond the scope of this paper, and and the relevant theory for neural networks is already done well in the original work of Cohen and Welling [3]. The purpose of this section is to establish the notation and framework for the rest of the paper, and it is mostly a reprisal of Cohen and Welling as well as of a blog post from Fuchs [8]. A group $G$ is a set of elements $g \in G$ together with a binary operation $\circ$ that combines two elements to form a third element, satisfying the following properties:

- Closure: For all $g_1, g_2 \in G$, the result of the operation $g_1 \circ g_2$ is also in $G$.
- Associativity: For all $g_1, g_2, g_3 \in G$, $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.
- Identity element: There exists an element $e \in G$ such that for every element $g \in G$, the equation $e \circ g = g \circ e = g$ holds.
- Inverse element: For each $g \in G$, there exists an element $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$.

A symmetry of an object is a transformation that leaves the object unchanged, with respect to some property. The set of all symmetries of an object forms a group, with function composition as the group operation, called the symmetry group of the object. The primary symmetry groups of interest in this paper are the rotation group $C_4$, the group of all discrete translations by pixels in 2D images $\mathbb{Z}^2$, and their combination, $p4$. A function $f : X \to Y$ is said to be equivariant with respect to a group $G$ if, for every transformation $g \in G$ and every input $x \in X$, the following condition holds:

$$f(g \cdot x) = g' \cdot f(x), \tag{1}$$

where $g \cdot x$ denotes the action of the group element $g$ on the input $x$, and $g' \cdot f(x)$ denotes the corresponding action of the group element $g'$ on the output $f(x)$. Note that equivariance is only defined up to the choice of group actions on input and output spaces. Invariance is a special case of equivariance where the group action on the output space is the identity transformation for all group elements, i.e., $f(g \cdot x) = f(x)$ for all $g \in G$ and $x \in X$. For a helpful illustration, image segmentation is often an equivariant task, whereas image classification is an invariant task.

### 2.1 Convolutions

To understand group equivariant CNNs in 2D, one needs to carefully define the difference between a pixel, an image, and a feature map. We will start by defining traditional convolutions on 2D images, and then extend the definition to group convolutions. A pixel is a single point in a 2D grid, identified by its coordinates $(x, y) \in \mathbb{Z}^2$, where $\mathbb{Z}$ is the set of integers. For notational convenience, we will refer to pixels using just one variable, e.g., $(x)$, with the understanding that it represents a vector. An image is a function $f : \mathbb{Z}^2 \to \mathbb{R}^K$ that maps each pixel to a $K$-dimensional vector, representing the color channels of the image. For the rest of this paper, we will consider grayscale images with $K = 1$ for simplicity. A kernel is another image $\psi : \mathbb{Z}^2 \to \mathbb{R}$ that is typically non-zero only in a small neighborhood around the origin, such as a $3 \times 3$ square. Note that all images are defined over the entire 2D grid, even though in practice they are only non-zero in a finite region. The correlation of an image $f$ with a kernel $\psi$ is a new image defined as:

$$(f * \psi)(t) = \sum_{x \in \mathbb{Z}^2} f(x)\psi(x - t), \tag{2}$$

where the sum is taken over all pixels in the 2D grid. The best way to understand this is as a two-step operation: translate the kernel by $t$, then compute the dot product between the translated kernel and the image, summing over all pixels. (Note that to transform the kernel, we apply the inverse translation to the pixel coordinates; see Cohen and Welling for details [3].) This new image can then be passed to another convolutional layer, and so on, forming a CNN.

The key conceptual step to extend to group convolutions is to recognize that the set of pixels $\mathbb{Z}^2$ is itself a group under addition, and is isomorphic to the group of discrete translations. Therefore, we can think of all images as defined on the group of *translations*. To generalize to an arbitrary group $G$, we redefine images as functions $f : G \to \mathbb{R}^K$ that map each group element to a $K$-dimensional vector.

$$(f * \psi)(t) = \sum_{g \in G} f(g)\psi(g^{-1}t), \tag{3}$$

where the sum is taken over all group elements $g \in G$. Note that for this definition to make sense, both the image $f$ and the kernel $\psi$ must be defined on the entire group $G$. For the first layer of a group equivariant CNN, the input image is still defined on the translation group $\mathbb{Z}^2$, so it must be lifted to the full group $G$ using a lifting convolution:

$$(f * \psi)(g) = \sum_{x \in \mathbb{Z}^2} f(x)\psi(g^{-1}x). \tag{4}$$

The full proof that group convolutions are equivariant to the group $G$ is given in Cohen and Welling [3], and mostly comes down to keeping track of transformations of images versus group elements. Here we give only an intuitive sketch: the convolution evaluated at $g$ depends only on the relative transformation of the kernel and the image. Transforming an image by some group element $h$ is equivalent to transforming the kernel by $h^{-1}$. The value of this new convolution at $g$ is therefore equal to the value of the original convolution at $h^{-1}g$, which is exactly the definition of equivariance.

## 3 Relaxed Equivariance

Wang et al. [7] propose several methods to relax the strict equivariance constraint of group equivariant CNNs, such as residual pathway priors [9] and combo nets, which use non-equivariant layers followed by equivariant layers. For this paper, we focus on the principal method of **Relaxed Group Convolutions** (RGCs). For each kernel $\psi$ in a group convolutional layer, they replace it with a linear

combination of $L$ kernels, weighted by a symmetry-breaking function $w(h)$ on the group $G$:

$$\psi_{\text{relaxed}}(g,h) = \sum_{l=1}^{L} w_l(h)\psi_l(g^{-1}h). \tag{5}$$

Group convolution is then defined as:

$$(f * \psi_{\text{relaxed}})(g) = \sum_{h \in G} f(h)\psi_{\text{relaxed}}(g,h). \tag{6}$$

The value at $g$ now depends on the specific combination of $g, h$, breaking the strict equivariance constraint. Equivariance is recovered exactly when $w_l(h)$ is constant for all $h \in G$, for all layers $l$. These $w_l$ are learnable parameters, so that the model can learn how much to deviate from equivariance based on the data. In order to encourage the model to stay close to equivariance, Wang et al. [7] also introduce a regularization term that penalizes deviations from constant $w_l$:

$$\mathcal{L}_{\text{reg}} = \alpha \sum_{l=1}^{L} \sum_{g,h \in G} \|w_l(g) - w_l(h)\|. \tag{7}$$

The hyperparameter $\alpha$ controls the strength of the regularization, with higher values encouraging more equivariance.

Wang et al. [7] test their model on fluid dynamics data, simulating 2D smoke flow over a non-uniform buoyancy field, as well as using experimental jet flow data. They show that their relaxed equivariant model outperforms both normal CNNs and strictly equivariant CNNs on this task. This is a good proof of concept for a physical system, but because it is so specific to fluid dynamics, it is difficult to interpret the results in a general way. In the next section, we outline our experimental methodology to test relaxed equivariance on a more general task.

# 4 Experimental Methodology

We propose a new experimental setup to test the effectiveness of relaxed equivariant CNNs on a more general task. Whereas Wang et al. [7] create non-equivariant data by perturbing the physical constraints of the system, we separate the problem from the realm of physical systems entirely. Inspired by the content of AM226, we take a signal-noise perspective and create perturbations to equivariant data by adding anisotropic noise. We start with a modified MNIST dataset rotated by random multiples of 90 degrees, which consists of images of handwritten digits, making the classification task rotation-equivariant. After rotation, we then implement two different noising schemes. The first adds isotropic Gaussian noise to each pixel, which preserves rotation equivariance in expectation. The second adds anisotropic Gaussian noise, where the variance of the noise depends on the pixel location in a non-rotation-equivariant way. The target function is then no longer equivariant: it should treat the signal equivariantly, but it must also see through the noise, which is a non-equivariant task, since certain pixels are more reliable than others. In other words, an anisotropically noised image that is then rotated or translated should not result in a correspondingly rotated output: if it did so, the model would be failing to use the meaningful orientation information about the noise distribution. By varying the level of anisotropic noise, we can control how far the target function deviates from equivariance. We then train three different models on this data: a normal CNN, a strictly rotation-equivariant CNN using group convolutions over the $p4$ group, and a relaxed $p4$-equivariant CNN using RGCs as described in Section 3. We test the models on both noised and clean test data, and compare their performance across different noise levels. We hypothesize that the relaxed equivariant model will outperform both the normal CNN and the strictly equivariant CNN on the anisotropic noise data, while the strictly equivariant CNN will perform best on the isotropic noise data.

The anisotropic noise is generated by creating a mapping from pixel location to noise strength, as shown in Figure 1. The particular mapping used here has a vertical half-wavelength variation in noise strength along the left-hand side, and a horizontal double-wavelength variation along the right-hand side. This creates two bands of lower noise on the right-hand side of the image. We then sample Gaussian noise for each pixel with variance proportional to the mapped noise strength at that pixel.

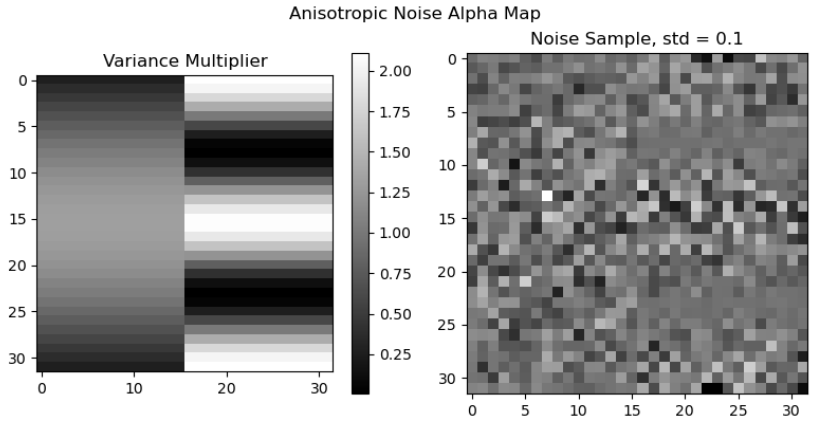



Figure 1: Left: strength of noise by pixel location for anisotropic noise. Right: sample of anisotropic noise on a blank image. The noise is weaker in two bands on the right-hand side.

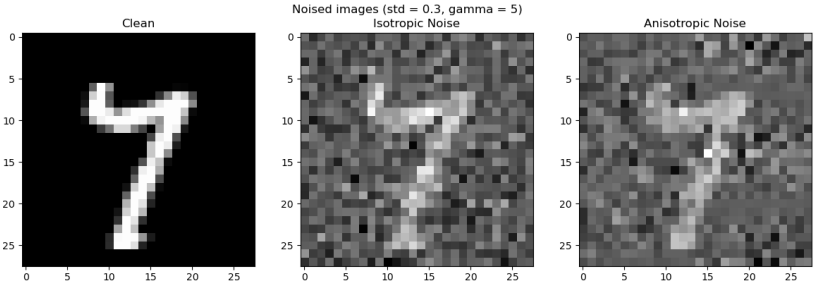
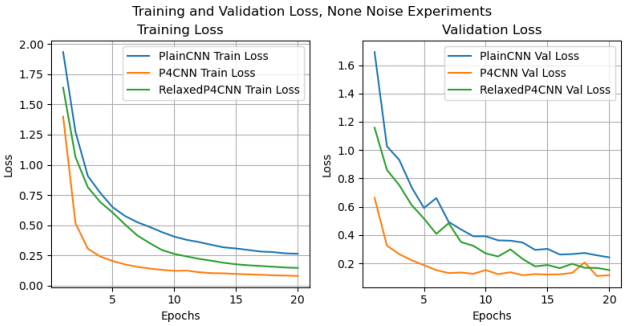



Figure 2: A sample image, unrotated, under the two noising schemes.

This particular mapping was chosen heuristically as something relatively easy to implement. In order to balance the noise between isotropic and anisotropic settings, we set the mean noise strength across all pixels to be equal. Equivalently, the trace of the covariance matrix of the noise is the same in both settings. For simplicity, the noise in all cases is uncorrelated between pixels.

Also for simplicity, we use only convolutional layers, with no pooling or batch normalization until the final layer, where we use global average pooling to ensure invariance, followed by a fully connected layer to output class probabilities.

The timeline of this project and limited access to computational resources prevented a comprehensive grid search over hyperparameters, especially considering that the relaxed equivariant model introduces the additional regularization hyperparameter $\alpha$. Therefore, model hyperparameters for the plain convolution network are based on the original work of Cohen and Welling [3], and the hyperparameters for the other models are chosen to be comparable. Of course, we also scale the number of hidden dimensions to keep the number of parameters roughly equal across models. Model hyperparameters can be found in Appendix A.1. We use the Adam optimizer with a learning rate of 0.002 and a batch size of 64. The regularization strength $\alpha$ for the relaxed equivariant model is set to 0.0001 based on preliminary experiments, and the primary loss function is cross-entropy loss. The total number of epochs is also limited to 20 due to computational constraints. All code for this project is implemented in PyTorch. The model architecture for the group equivariant and relaxed equivariant CNNs is modified from a tutorial accompanying the paper from Wang et al. [10].




(a) No noise

Figure 3: Training and validation accuracy with no added noise.

## 5 Results

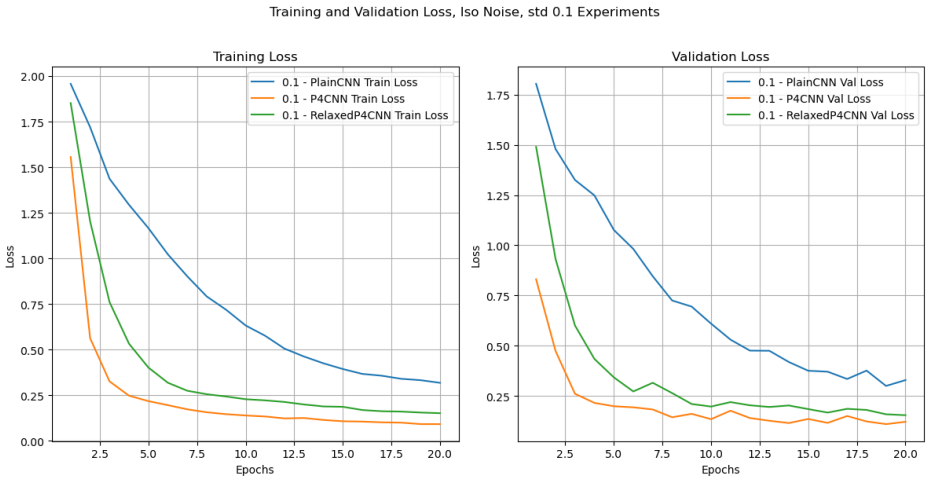
Table 1: Test accuracies (%) across noise types and levels. Columns show clean vs. noisy test accuracy for each model.

| **Noise** | $\sigma$ | PlainCNN | | P4CNN | | RelaxedP4CNN | |
|---|---|---|---|---|---|---|---|
| | | Noisy | Clean | Noisy | Clean | Noisy | Clean |
| Aniso | 0.1 | 92.35 | 89.97 | 96.26 | 93.16 | 93.61 | 88.87 |
| Aniso | 0.2 | 87.72 | 87.97 | 95.97 | 93.84 | 92.90 | 89.90 |
| Aniso | 0.3 | 88.85 | 86.31 | 94.05 | 92.22 | 91.64 | 75.74 |
| Iso | 0.1 | 90.06 | 90.52 | 96.53 | 53.59 | 95.46 | 95.45 |
| Iso | 0.2 | 91.05 | 91.61 | 96.01 | 96.08 | 90.95 | 87.63 |
| Iso | 0.3 | 88.79 | 90.37 | 94.36 | 75.46 | 94.44 | 93.91 |
| None | 0.1 | 92.86 | 92.83 | 96.65 | 96.65 | 95.23 | 95.35 |

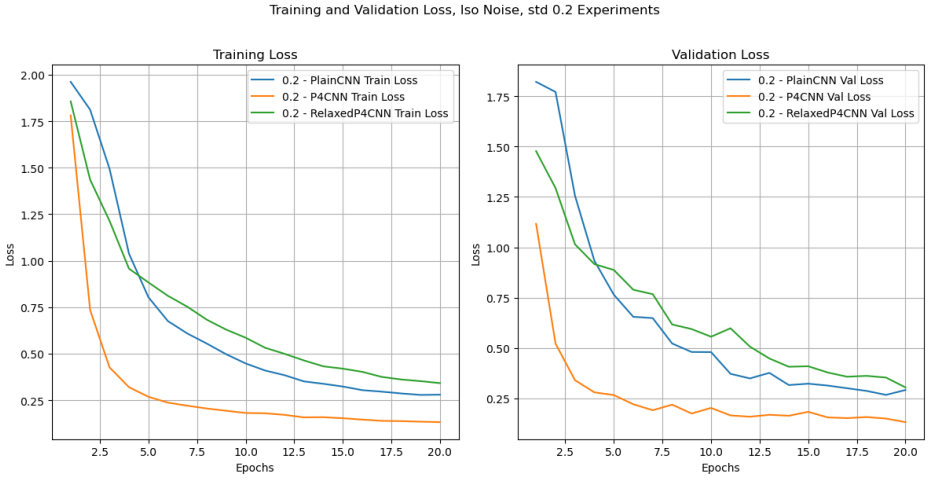
We summarize results in Table 1, showing test accuracy on both noised and clean data for different noise levels. We also plot training dynamics for all model runs, showing training and validation accuracy side by side for easy comparison.

## 6 Discussion and Conclusion

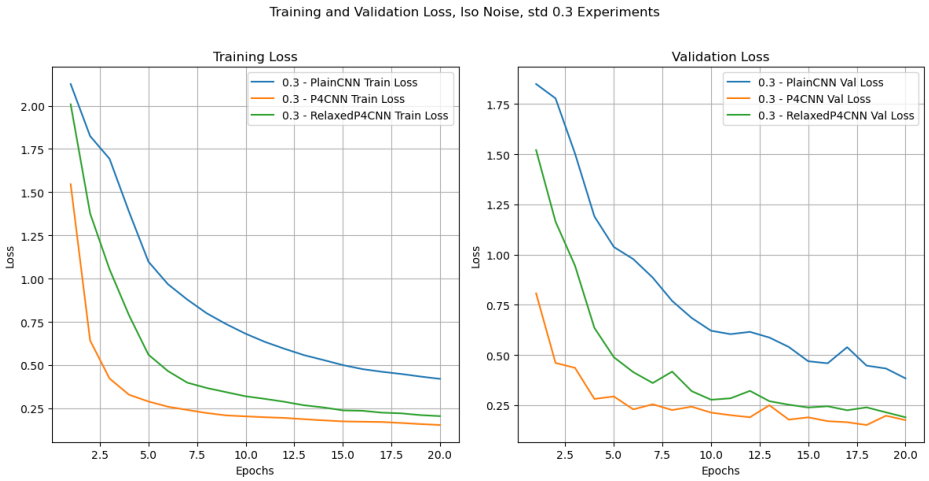
We have presented the underlying mathematics of fully and approximately equivariant CNNs, and conducted an experiment to investigate the relative performance of different models in a signal-noise setting, where the underlying signal is equivariant but the noise is not. Contrary to Wang et al. [7] and to our original hypothesis, the fully equivariant model outperforms all other models in all noise settings, and actually outperforms relaxed equivariance by a greater margin with anisotropic noise than with isotropic noise. This is surprising, because we expect that models that are "aware" of the orientation of the noise will perform better. There are a couple of possible explanations for this result. One possibility is that the noise is not sufficiently anisotropic, so the costs of deviating from equivariance outweigh the benefits: The relaxed model has significantly fewer hidden layers per convolutional layer to keep the parameter count similar, which may limit its expressivity. Additionally, the fully equivariant model trains faster, likely because it has a smaller search space. Our runs were cut off before full convergence, so the differences we see may be representations of the training dynamics, rather than the final performance. Furthermore, the hyperparameter search was limited, and inferred from the Cohen and Welling [3] paper, which was optimal for the fully equivariant model, but may not be optimal for the other models. This is supported by the poorer performance of the relaxed model on clean data. In order to further investigate these possibilities, a more comprehensive hyperparameter search and longer training times would be necessary. It would also be good to test a wider variety of anisotropic noise patterns, to see if the results are consistent across different types of anisotropy, including correlated noise.




(a) Isotropic — low

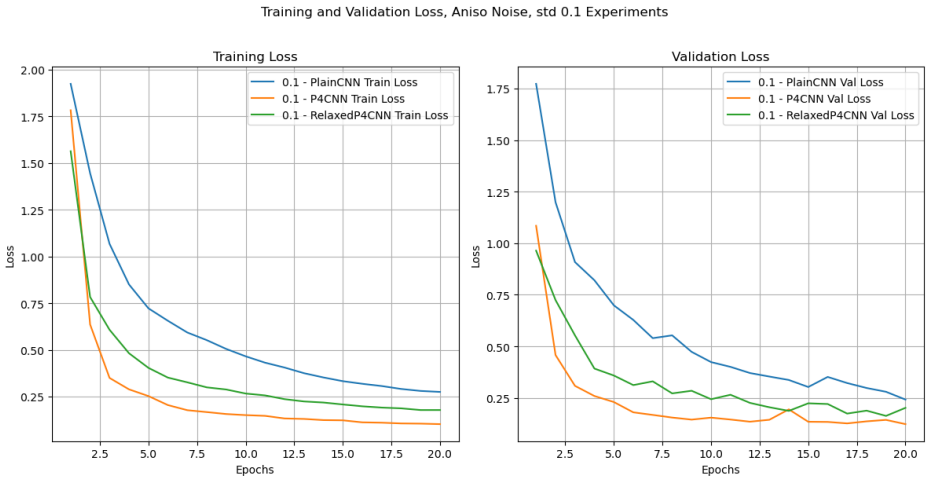



(b) Isotropic — medium

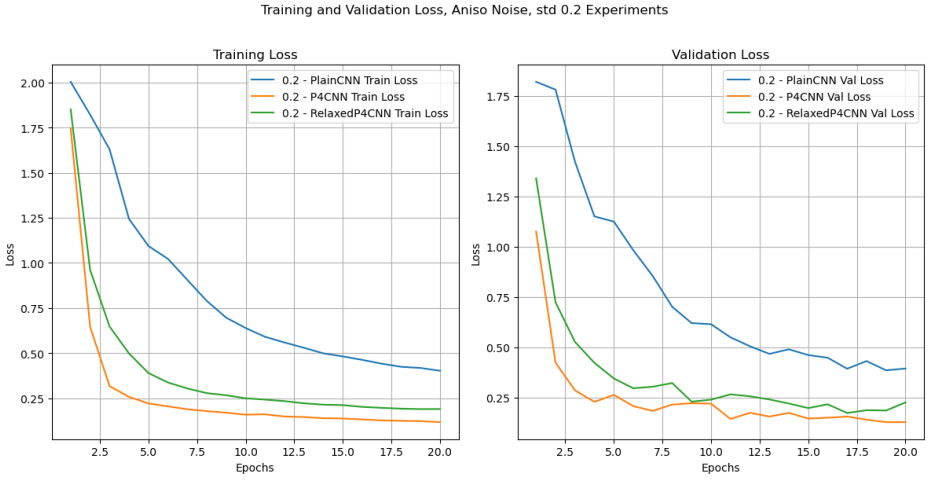



(c) Isotropic — high

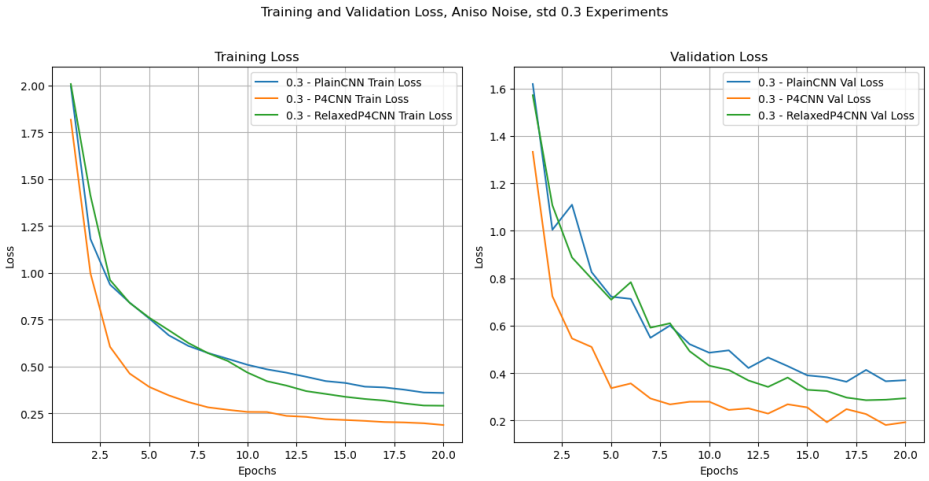
Figure 4: Training and validation accuracy for all isotropic noise levels.




(a) Anisotropic — low




(b) Anisotropic — medium




(c) Anisotropic — high

Figure 5: Training and validation accuracy for all anisotropic noise levels.

# References


[1] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL https://doi.org/10.1007/BF02551274.

[2] Richard S. Sutton. The bitter lesson. 2019. Accessed: 2025-12-13.

[3] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *CoRR*, abs/1602.07576, 2016. URL http://arxiv.org/abs/1602.07576.

[4] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, Apr 1980. ISSN 1432-0770. doi: 10.1007/BF00344251. URL https://doi.org/10.1007/BF00344251.

[5] Nathaniel Thomas, Tess E. Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://arxiv.org/abs/1802.08219.

[6] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13 (1):2453, May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL https://doi.org/10.1038/s41467-022-29939-5.

[7] Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. *CoRR*, abs/2201.11969, 2022. URL https://arxiv.org/abs/2201.11969.

[8] Ed Wagstaff and Fabian Fuchs. Cnns and equivariance – part 1/2. https://fabianfuchsml.github.io/equivariance1of2/, 2024. Accessed: 2025-12-14.

[9] Marc Finzi, Gregory W. Benton, and Andrew Gordon Wilson. Residual pathway priors for soft equivariance constraints. *CoRR*, abs/2112.01388, 2021. URL https://arxiv.org/abs/2112.01388.

[10] Rui Wang, Robin Walters, and Rose Yu. Approximately-equivariant-nets. https://github.com/Rose-STL-Lab/Approximately-Equivariant-Nets, 2022. GitHub repository.


# A Appendix

## A.1 Model Hyperparameters

All models have a total of 7 convolutional layers, with ReLU activations in between, followed by global average pooling and a fully connected layer to output class probabilities. The plain CNN has 20 hidden channels in each convolutional layer, while the equivariant CNN has 10 hidden channels, and the relaxed equivariant CNN has 6. Following Wang et al. [7], the relaxed equivariant model uses $L = 3$ kernels per layer for the RGCs. Parameter counts are summarized in Table 2.

| Model | Number of Parameters |
|---|---|
| Plain CNN | 25,750 |
| Equivariant CNN | 25,400 |
| Relaxed Equivariant CNN | 27,544 |

Table 2: Number of parameters for each model.