Evaluating Machine Learning Models with NERO: Non-Equivariance Revealed on Orbits

Zhuokai Zhao[1], **Takumi Matsuzawa**[2], **William Irvine**[2], **Michael Maire**[1], **Gordon Kindlmann**[1]
[1]Department of Computer Science, University of Chicago,
[2]James Franck Institute and Department of Physics, University of Chicago
Correspondence to zhuokai@uchicago.edu

Abstract

Traditional scalar-based error metrics, while quick for assessing machine learning (ML) model performance, often fail to expose weaknesses or offer fair evaluations, particularly with limited test data. To address this growing issue, we introduce *Non-Equivariance Revealed on Orbits* (NERO), a novel evaluation procedure that enhances model analysis through assessing equivariance and robustness. NERO combines a task-agnostic interactive interface with a suite of visualizations to deeply analyze and improve model interpretability. We validate the effectiveness of NERO across various applications, including 2D digit recognition, object detection, particle image velocimetry (PIV), and 3D point cloud classification. Our case studies demonstrate the ability of NERO to clearly depict model equivariance and provide detailed insights into model outputs. Additionally, we introduce *consensus* as an alternative to traditional ground truths, expanding NERO to unlabeled datasets and enabling broader applications in diverse ML contexts.

1 Introduction

Machine learning (ML) has significantly advanced various research fields [Voulodimos et al., 2018, Kelleher, 2019]. The evaluation process in ML is unfortunately largely unchanged, hindering interpretation and further innovation. Model quality is typically measured with a scalar, such as accuracy for classification tasks, precision and recall for object detection, and mean squared error for more quantitative tasks. Comparing models via scalar metrics can miss important details, limiting insight for ML researchers, and creating uncertainty for practitioners. Two models can be quantitatively similar on average, but respond very differently to meaningfully changing individual inputs. For example, Fig. 1 illustrates two models for finding humans crossing streets. A model that responds erratically to translating the field of view (which should only translate the predicted bounding box) may be less trustworthy even if it performs better *on average* on a fixed test set.

Empirical science is especially challenging for applied ML. Specialized instrumentation means data is expensive to gather and labor-intensive to label. Popular ML models, however, need large training and testing datasets, in part due to the simplicity of their scalar metric evaluations. For example, the popular Microsoft COCO [Lin et al., 2015] dataset for object detection has 328,000 labeled images; Object365 [Shao et al., 2019] has over 2 million. The ubiquity of ML for object detection justifies and amortizes the cost of creating such datasets, but this scaling does not generally apply to experimental science. Also, scientists value robustness, predictability and interpretability in their computational tools [Oviedo et al., 2022], unlike the black-box nature of deep learning. These issues have catalyzed research in interpretable machine learning (IML) [Molnar, 2020, Saranya and Subhashini, 2023], which seeks to reveal ingredients of model predictions. Appendix A reviews related work.

Our work complements IML research by depicting ML model response in terms of *equivariance*, a mathematically grounded way to relate changes in model inputs to changes in outputs.

Interpretable AI: Past, Present and Future Workshop at NeurIPS 2024

In Fig. 1, for example, translating the input image should consistently correspond to translations of the output bounding box. We organize our visualization of model equivariance around a mathematical *group* of input transformations and the set of all transformations (the *orbit*) of a given input. This is captured in our proposed *Non-Equivariance Revealed on Orbits (NERO) Evaluation*, which shows how equivariant a model is, and the structure of its equivariance failures. In settings where practitioners can reason about their analysis task in terms of mathematically predictable responses to data transforms, NERO evaluation gives an informative and detailed picture of ML model performance that scalar summary metrics elide. NERO provides a data-efficient way of testing ML models, making thorough and fair evaluations possible without the acquisitions of large datasets.

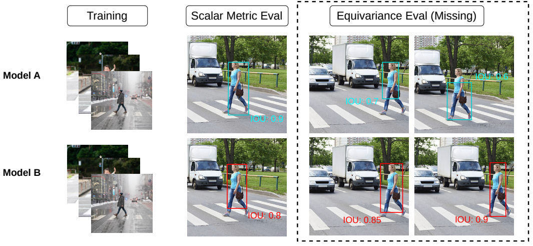


Figure 1: Scalar metric evaluations can be misleading, seen here with models A and B for pedestrian localization. Despite Model A on average outperforming B with Intersection Over Union (IOU), it fails in critical edge cases where B excels, showing the need for more nuanced assessments.

The contributions of this paper, are (1) NERO Evaluation, an integrated workflow that visualizes model equivariance in an interactive interface to facilitate ML model testing, troubleshooting, evaluation, comparison, and to provide better interpretation of model behaviors; and (2) *consensus*, a proxy for ground truth that helps evaluate model equivariance with unlabeled data.

2 Methodology

2.1 Overview

Appendix B outlines mathematical basics of groups, group actions. We say generally that a model is equivariant if it responds to a change in input with a correctly corresponding change in output (the intent illustrated in Fig. 1). Real models often fall short of this; and NERO evaluations visualize *how*. Fig. 2 defines the NERO plot in terms of how a change from x to x' in input space X corresponds to a change from y to y' in output space Y. The NERO plot visualizes the *gap* between $h(x')$ (the model applied to the transformed input) and y' (the correspondingly transformed ground truth y). This abstractly depicts group G to schematically indicate the orbits $G(x)$ of x and $G(y)$ of y, but some particular spatial layout of G necessarily determines the shape of the NERO plot. *If the model is equivariant, then $h(x') = y'$, so the* NERO *plot is a flat constant.* The visual structure of a non-constant NERO plot reveals the model non-equivariance over the group orbit. The quantity shown in a NERO plot is any scalar metric (understandable to practitioners) that measures the gap between $h(x')$ and y', including prediction confidence, accuracy, mean square error (MSE), or other error metrics. The NERO plot illustrated in Fig. 2 (right) is an **individual NERO plot** (§2.3), as it depicts model non-equivariance along the group orbit $G(x)$ around an individual input sample x.

While §1 critiqued single scalars to summarize model results over a large dataset, informative NERO plots can also involve averaging. An **aggregate NERO plot** (§2.2) visualizes the average scalar metric over a dataset, or a subset of it, at each point along the group orbit (i.e. with the same domain as the individual NERO plot), to show trends in the model's response to transformed inputs. Like PDP and ICE plots (§A.2), aggregate NERO plots evaluate the model within some neighborhood around a

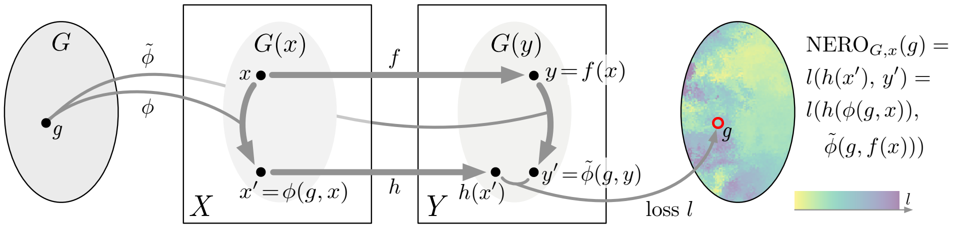
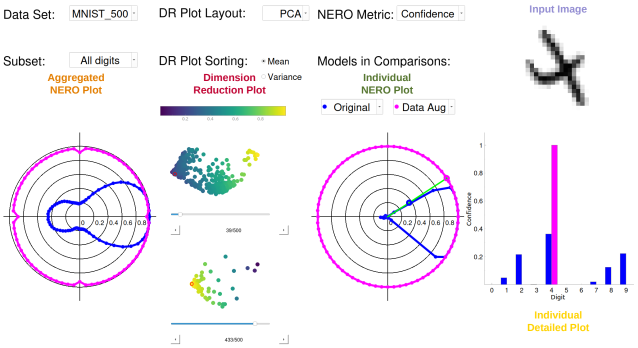


Figure 2: Overview of NERO. An ML model has inputs X and outputs Y. G is a transformation group acting on X with ϕ, and on Y with $\tilde{\phi}$. The group element $g \in G$ transforms x to $x' = \phi(g, x)$; the set of all possible transforms is the orbit $G(x)$. The ground truth $y = f(x) \in Y$ is transformed by g to $y' = \tilde{\phi}(g, y)$, which serves as ground truth to evaluate (here with loss function l) the result $h(x')$ of evaluating the model on transformed input x'. $\mathrm{NERO}_{G,x}$ visualizes loss over the orbit $G(x)$.

given sample, but instead of varying features in isolation, we traverse the orbit of some interpretable transform group. To try to see degrees of freedom lost in the aggregate NERO plot, we can also treat the individual NERO plots as n-vectors, and use dimensionality reduction. The resulting **dimension reduction (DR) scatter plot** (§2.4) organizes data points according to the similarity of their patterns of non-equivariance, to help localize abnormal model behavior and identify the connections between worse-performing cases. All of these visualizations are linked together in the interactive **NERO interface** (Appendix C) that provides users with both the convenience to see model equivariance in a high-level view across a whole dataset through the aggregate NERO plot, and navigating into details through the individual NERO plot, e.g., a specific place in the orbit where the model has trouble.

The following subsections illustrate the components of NERO Evaluation through a digit recognition task on MNIST [Lecun et al., 1998], with the group action being continuous rotations around the image center. More specifically, NERO Evaluation is presented via an interactive NERO interface; Fig. 3 shows an example. We use this task and the MNIST dataset to first concretely illustrate NERO evaluation in a well-known setting, not because this task exemplifies the scientific tasks for which we created NERO. §3 showcases more representative tasks and applications.


Data Set: MNIST_500

DR Plot Layout: PCA

NERO Metric: Confidence

Subset: All digits

DR Plot Sorting: ● Mean ○ Variance

Models in Comparisons: ● Original ● Data Aug

Figure 3: The five sections of the NERO interface are (left to right): aggregate NERO plot, dimension reduction (DR) plot, individual NERO plot, input image, and individual detailed plot. Each section name is labeled here only for illustration purposes. The sections are interactively controlled, with linked views.

For the proposed NERO evaluation to be effective, the first criterion is to ensure that the associated NERO plots are distinguishable enough when evaluated on two models with different equivariance. To illustrate how NERO plots differ on equivariant and non-equivariant models, a simple neural network consisting of six cascaded convolutional layers with batch normalization [Ioffe and Szegedy, 2015] and ReLU [Glorot et al., 2011] is purposefully trained twice, first without and then with rotational augmentation, to create two models that differ predictably. That is, the data augmented (DA) model should have better invariance, although the total amount of training with or without rotation augmentation is the same. Notably, the reason why using data augmentations to generate different ML models is not to prove the effectiveness of data augmentation, but to generate models with clear, controllable behaviors so that the correctness as well as expected behavior from NERO can be verified.

2.2 Aggregate NERO Plot

Aggregate NERO plots provide an overview of equivariance across a dataset, using consistent visual encoding as in individual plots (more details in §2.3). These plots, such as those shown in Fig. 4, demonstrate rotational symmetry for each digit in the MNIST dataset. For example, digit “0” exhibits rotational invariance, while “1” shows symmetry at 0° and 180°, and digits like “2” and “3” lack symmetry. The DA model (magenta) is more equivariant than the original (blue), indicating smoother, circular NERO plots. Although NERO plots do not explain model predictions, they highlight patterns in model behavior.

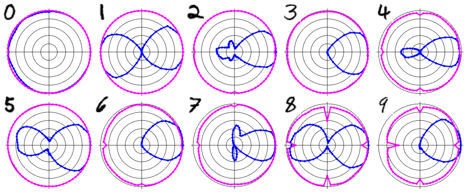


Figure 4: Aggregate NERO plots for the original (blue) model reflect the average rotational symmetry of each digit.

2.3 Individual NERO Plot

Individual NERO plots (Fig. 3 third from left) visualize model equivariance for a single sample, using confidence as the metric (probability of correct classification). These plots show confidence over rotation angle θ; a perfect circle indicates perfect rotational equivariance, while dips reveal non-equivariance. As expected, the DA model (magenta) shows near-perfect equivariance, while the original model (blue) has higher confidence at small angles. A bar chart displays model confidences for all digits at a selected angle, showing the DA model's higher accuracy for "4" compared to the original model. Individual NERO plots offer insights into data structure and task performance, such as for digits "6" and "9" (Fig. 5). Both the original and DA models perform similarly at small rotation angles, but the original model's performance drops significantly at larger angles, leading to mis-classifications (e.g., confusing a rotated "6" with a "9"). The DA model, while not perfectly equivariant like with digit "4", still maintains better performance across all angles, correctly identifying rotated "6"s and "9"s with moderate confidence. These plots, combined with detail views, help visualize model equivariance and interpret model behavior, revealing that even rotated digits in MNIST can be distinguished due to their unique shapes. Further examples in §3 will demonstrate these capabilities.

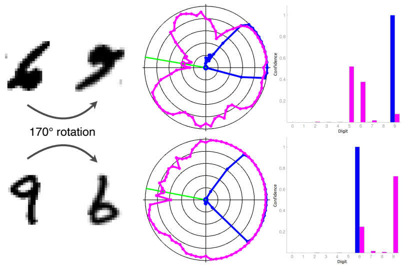


Figure 5: Individual NERO and detail plots of original (blue) and DA (magenta) models.

2.4 Dimension Reduction (DR) Plot

Conceptually, DR plots bridge the aggregate and individual NERO plots they are positioned between in the NERO interface (Fig. 3, second column from the left). By applying dimensionality reduction to the high-dimensional data vector underlying individual NERO plots, the DR plot lays out data points in a 2D scatterplot. Similar patterns of non-equivariance appear near each other, providing an overview of model performance and highlighting outliers. The scatterplot dots are color-coded by the mean or variance of the individual NERO plot values, aiding in trend identification or pinpointing extremes in equivariance. Users can interactively click on a dot to view corresponding individual and detail plots, as shown by the red circle in Fig. 3. Fig. 6 offers an expanded view, demonstrating that nearby points have similar non-equivariance patterns, while distant points show distinct patterns, even if their plots are similarly shaped but differ in orientation.

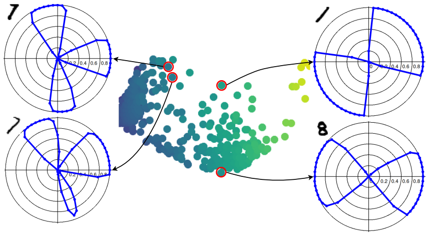


Figure 6: DR plot color-mapped by mean confidence, annotated with associated individual NERO plots, with input digits shown at top-left corners.

2.5 Consensus

Our intent with using existing scalar metrics (accuracy, confidence) for making NERO plots is to ease their interpretation and adoption by practitioners. However, NERO can in principle also apply to unlabeled data, since (Fig. 2) equivariance is revealed through the gap between $h(x')$ and y', which need not depend on having ground truth. However, given that existing metrics all require ground truth, an additional modest contribution of this work is *consensus*, which serves as a proxy for ground truth in the metric evaluation, when making NERO evaluations to assess model equivariance or covariance (as opposed to invariance). The consensus for input x is roughly the average of the untransformed model outputs on all transformed inputs within the orbit. Relative to Fig. 2, we have

$$\text{consensus}(x) = \left\langle \tilde{\phi}(g^{-1}, h(\phi(g, x))) \right\rangle_{g \in G} \tag{1}$$

The average $\langle \cdot \rangle_G$ depends on the structure of output space Y, while G depends on the equivariance of interest. For object detection, Y is the set of bounding boxes defined by corners, and an element (t_x, t_y) of translation group G acts on the bounding box by component-wise addition. In this case, Eq. (1) can be computed by simple arithmetic mean of the translated bounding box corners.

3 Experimental Case Studies

As previously stated, NERO evaluation is model- and task-agnostic. This section showcases its application across different research areas: object detection in 2D photographic images (§3.1), velocity measurements in fluid dynamics via particle image velocimetry (§3.2), and classification in 3D computer vision with point clouds (§3.3). Additionally, we evaluate *consensus* in §3.4.

3.1 Object Detection

As shown in Fig. 1, scalar-metric evaluations do not directly address corner-case performance in object detection, which is crucial for applications like autonomous driving. This section demonstrates how NERO offers a superior evaluation pipeline using Faster R-CNN [Ren et al., 2015] and MSCOCO [Lin et al., 2015], though NERO's application is model-agnostic.

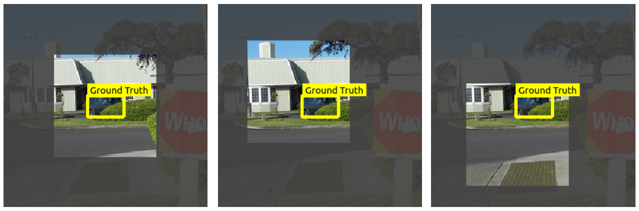
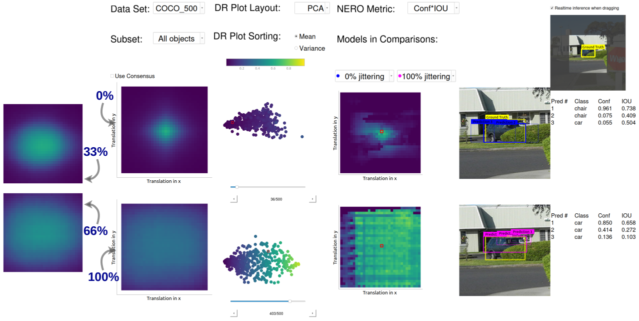


Figure 7: Key objects are shifted by cropping the original MSCOCO image to shifted bounds (the non-masked square).


Data Set: COCO_500
DR Plot Layout: PCA
NERO Metric: Conf*IOU
☑ Realtime inference when dragging
Subset: All objects
DR Plot Sorting: ● Mean ○ Variance
Models in Comparisons: ● 0% jittering ● 100% jittering
□ Use Consensus

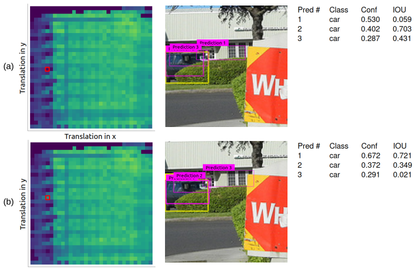
Pred #	Class	Conf	IOU
1	chair	0.961	0.738
2	chair	0.075	0.409
3	car	0.055	0.504

Pred #	Class	Conf	IOU
1	car	0.850	0.658
2	car	0.414	0.272
3	car	0.136	0.103

Figure 8: NERO interface for object detection, for models trained with 0% (upper row) and 100% (lower row) jittering. Sections for aggregate, dimension reduction, individual, and detail plots are organized as in the MNIST interface (Fig. 3). Two aggregate NERO plots on left edge show intermediate jittering levels for comparison.

The architecture of Faster R-CNN does not guarantee translational equivariance, so models with different equivariance properties can be obtained by training with datasets with different augmentations, as we show here. We selected 5 out of the 80 MSCOCO classes for demonstration: *car*, *bottle*, *cup*, *chair* and *book*. We selected objects that belong to these 5 classes as key objects and cropped the original images to a 128×128 window around these objects. As showed in Fig. 7, translational shifts (by between -64 and 64 pixels in both directions) are achieved by cropping with shifted bounds, so that the key object positions change within the field of view. To ensure interesting cropped images, the MSCOCO images are filtered with following criteria: (1) include a key object whose ground truth class label is in the 5 selected classes; (2) ensure that for all shifts the cropped fields of view does not extend past the original image edges; and (3) ensure that the key object's ground truth bounding box is not less than 1% or more than 50% of the cropped 128×128 region.

We hypothesize that varying levels of model equivariance can be induced by adjusting random shifts, or "jittering", during training. At 0% jittering, key objects remain centered in cropped images, whereas at 100% jittering, objects are randomly shifted within a $[-64, 64]$ range. Models trained with 0% jittering are expected to perform well only on unshifted images, while those trained with 100% jittering should exhibit higher equivariance.


Pred #	Class	Conf	IOU
1	car	0.530	0.059
2	car	0.402	0.703
3	car	0.287	0.431

Pred #	Class	Conf	IOU
1	car	0.672	0.721
2	car	0.372	0.349
3	car	0.291	0.021

Figure 9: Individual NERO plots examine a 100% jittering model: (a) explores a dark spot, while (b) analyzes a nearby spot with better results.

Fig. 8 displays the complete NERO interface for models trained with 0% and 100% jittering. Similar to the MNIST example (Fig. 3), model equivariance is evaluated using both aggregate and individual NERO plots, connected via DR plots, with task-specific detail plots on the right. The left edge of Fig. 8 also includes aggregate NERO plots for two intermediate jittering levels. As expected, jittering levels correlate visually with the width of the NERO plot peak, indicating high equivariance in the 100% jittering model (lower row) and non-equivariance in the 0% jittering model (upper row).

While aggregate NERO plots provide a quick overview of model equivariance, individual NERO plots allow for deeper investigation. For instance, in Fig. 8, the 100% jittering model's individual NERO plot reveals dark regions on the left edge, suggesting poorer performance at specific shifts. Practitioners can click on these spots for detailed analysis, as demonstrated in Fig. 9, which explores a small shift change in the same model. At both shifts, the model produces three bounding box predictions with a high IOU of about 0.7, but the confidence ranking differs between the two locations. The individual NERO plot focuses on the IOU for the most-confident prediction, highlighting dark regions that guide practitioners in exploring and understanding model edge cases.

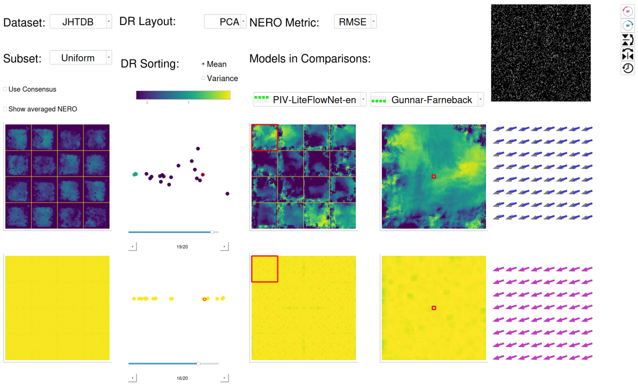
User study. A researcher with knowledge in both computer vision and equivariant ML, tried our NERO evaluation for object detection. The evaluation was semi-guided, meaning that the expert was free to explore himself after we walked him through examples similar to those earlier in this section. The ensuing discussion focused on the NERO plot idea itself and its value; quotes below from the expert are in italics. *It is intuitive to present equivariance with simple group theories* – the expert understood how we transform samples along group orbits, and measure results on transformed samples. *Aggregate NERO plots are quick to look at when comparing two models* – the expert felt that NERO plots do not create excessive visual complexity for users. *clicking on these dots to locate single samples is very helpful ...* – the expert said about the DR plots – *... now I can see what are the reasons behind the different performance* – the expert looking at the corresponding individual and detail plots. After using the interface for about 10 minutes, the expert concluded: *Using equivariance as an evaluation strategy is interesting. Everyone knows there is more going on underneath the average errors we see everyday, but we are not able to easily, systematically capture and compare them until using NERO. I think NERO would benefit anyone who cares about model equivariance or develops better ENN.*

3.2 Particle Image Velocimetry (PIV)

Particle Image Velocimetry (PIV) is essential for studying fluid dynamics, using video frames of particles to estimate velocity flow fields [Raffel et al., 2018]. While traditional PIV algorithms handle simple flows, ML-based methods promise faster and more complex computations. However, thorough evaluation beyond metrics like RMSE is needed to trust the ML models. This section demonstrates how NERO can effectively evaluate equivariance in ML applications, comparing the traditional Gunnar-Farneback [Farnebäck, 2003] with the ML-based PIV-LiteFlowNet-en [Cai et al., 2019].

In total, 8,794 pairs of images covering 6 different types of flows, namely *Uniform*, *Backstep*, *Cylinder*, *SQG*, *DNS*, and *Isotropic*, are used during training. 120 image pairs are used in testing when generating the NERO plots. As expected, Gunnar-Farneback shows near-perfect equivariance, while PIV-LiteFlowNet-en has less. NERO plots (Fig. 10) highlight these differences, with detail plots allowing deeper investigation into model outputs.

User study. A physicist with expertise in PIV tried our NERO PIV interface and gave qualitative feedback. We followed the same procedure as in §3.1. *It is very good to see so much more information than an average value, ..., for a turbulence flow the interesting and hard part is not everywhere, often much less than the boring part, so the average error really does not help much.* – the expert likes that NERO plots show richer information than conventional scalar metrics. *Being able to locate high-variance (less-equivariant) samples from the DR plot is great* – the expert said when looking at the DR plots – *it is important to bring out the actual interesting samples to investigate* – the expert thinks the design is effective in helping user traverse through samples and locate the interesting one. *Yes, definitely, NERO would save me so much time analyzing PIV model outputs.* – the expert said when asked about if he would personally use the evaluation method in his research.



Dataset: JHTDB

DR Layout: PCA

NERO Metric: RMSE

Subset: Uniform

DR Sorting: ● Mean ○ Variance

Models in Comparisons: PIV-LiteFlowNet-en, Gunnar-Farneback

□ Use Consensus

□ Show averaged NERO

Figure 10: The NERO interface for PIV comparing an ML method (top) with a non-ML method (bottom).

3.3 3D Point Cloud Classification

Point cloud classification assigns semantic labels to 3D point clouds, with a focus on addressing performance issues from object rotations through equivariant models. NERO is used here to evaluate these models. Rotations are visualized in 2D NERO plots using a subset of rotations represented in axis-angle form within three 2D slicing planes. The Point Transformer model, which is invariant to point cloud permutations, was trained with and without rotation augmentation for comparison.

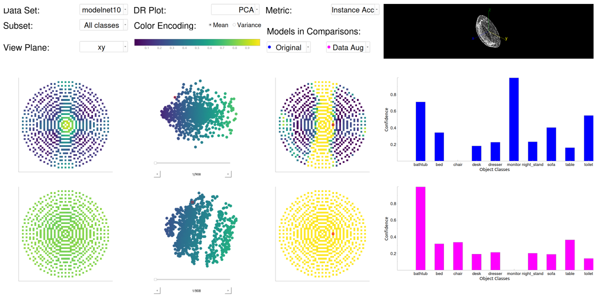

Data Set: modelnet10
Subset: All classes
View Plane: xy
DR Plot: PCA
Color Encoding: ● Mean ○ Variance
Metric: Instance Acc
Models in Comparisons: ● Original ● Data Aug

Figure 11: NERO interface for 3D point cloud classification comparing Point Transformer model trained without (top row) and with (bottom row) rotation augmentations.

We utilize the ModelNet10 subset from the widely recognized ModelNet40 dataset [Wu et al., 2015]. More specifically, the ModelNet40 dataset comprises 12,311 CAD models across 40 categories, divided into 9,843 training and 2,468 testing samples. ModelNet10, a subset of ModelNet40, includes 10 categories with 3,991 training and 908 testing samples. This data preparation process involves uniformly sampling point clouds from the CAD models as described by Qi et al. [2017]. As shown in Fig. 11, the original model performs well only with small rotations, while the augmented model shows greater invariance across all rotations. Individual NERO plots further illustrate these differences, such as in the case of a bathtub, where the augmented model consistently recognizes the object across all rotations.

User study. We invited the same expert from §3.1 to give us evaluations again. This time, we focused more on collecting how it feels going from one interface (application) to another. *It feels very similar, I am still able to quickly navigate myself to the places I am interested in* – the expert agrees that the similar high-level interface design successfully helps researchers quickly adapt from one application to another – *it is showing evaluation results way beyond scalar metrics, which could be very useful when evaluating and debugging model behaviors* – the expert agrees again that NERO evaluation provides more thorough and informative results than standard scalar metrics.

3.4 Consensus Evaluation

We evaluate the proposed *consensus* within an object detection scenario. In this context, the *consensus* of Eq. (1) represents the average of unshifted bounding box predictions, derived from shifted images. Fig. 12 illustrates the individual NERO plots generated from both the ground truth and the consensus. The notable similarity between these two plots suggests that the degree and structure of equivariance exhibited by the NERO plots are comparably effective with or without ground truth, suggesting that *consensus* enhances the utility of NERO plots when ground truth labels are not available.

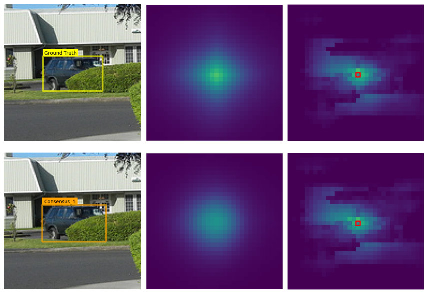


Figure 12: Consensus boxes (left), NERO plots from ground truth (middle), and from consensus (right).

4 Conclusions and Future Work

NERO represents a novel, interactive ML evaluation system that is built on model equivariance and basic group theory to address the inadequacies of evaluating ML models with scalar metrics. The examples we have showed in Section 2, 3.1 , 3.2, and 3.3 demonstrate four settings where NERO evaluations better assess model performance by revealing model equivariance and making black-box models more interpretable. In principle, the idea of using aggregate, dimension reduction, and individual NERO plots, linked in an interactive interface, extends natively to many other areas of ML research as well, facilitating findings and explorations of various model behaviors. In future, we plan to further study the idea of *consensus* (§2.5, Fig. 12), as it potentially frees ML evaluation from needing ground truth. And, as we briefly mentioned in §A.2, NERO plots can be a drop-in replacement for the conventional scalar-based evaluations that are widely employed in current ENN and surrogate model studies. Last but not least, we believe NERO could also be applied and improved to better assess model behaviors under adversarial attacks, which have been empirically proved to be devastating to neural networks performance. Other future work may also include conducting a more thorough user survey, and making NERO evaluation a standard library for common ML applications.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv:1603.04467 [cs]*, March 2016.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data Augmentation Generative Adversarial Networks. *arXiv:1711.04340 [cs, stat]*, March 2018.

Daniel W. Apley and Jingyu Zhu. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *arXiv:1612.08468 [stat]*, August 2019.

Serge Assaad, Carlton Downey, Rami Al-Rfou, Nigamaa Nayakanti, and Ben Sapp. Vn-transformer: Rotation-equivariant attention for vector neurons. *arXiv preprint arXiv:2206.04176*, 2022.

Shengze Cai, Jiaming Liang, Qi Gao, Chao Xu, and Runjie Wei. Particle image velocimetry based on a deep learning motion estimator. *IEEE Transactions on Instrumentation and Measurement*, 69(6): 3538–3554, 2019.

Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. Visualizing the Feature Importance for Black Box Models. *arXiv:1804.06620 [cs, stat]*, 11051:655–670, 2019. doi: 10.1007/978-3-030-10925-7_40.

Anadi Chaman and Ivan Dokmanić. Truly shift-invariant convolutional neural networks. *arXiv:2011.14214 [cs]*, March 2021.

Angelos Chatzimparmpas, Rafael M. Martins, Ilir Jusufi, and Andreas Kerren. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization*, 19 (3):207–233, July 2020. ISSN 1473-8716. doi: 10.1177/1473871620904671.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A Group-Theoretic Framework for Data Augmentation. *arXiv:1907.10905 [cs, math, stat]*, November 2020.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data. *arXiv:1805.09501 [cs, stat]*, April 2019.

Peijian Ding, Davit Soselia, Thomas Armstrong, Jiahao Su, and Furong Huang. Reviving shift equivariance in vision transformers. *arXiv preprint arXiv:2306.07470*, 2023.

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*, March 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the Landscape of Spatial Robustness. *arXiv:1712.02779 [cs, stat]*, September 2019.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer, 2003.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, October 2001. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1013203451.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, June 2011.

Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *arXiv:1309.6392 [stat]*, March 2014.

Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *arXiv:1801.06889 [cs, stat]*, May 2018.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, March 2015.

Minsuk Kahng, Pierre Y. Andrews, Aditya Kalro, and Duen Horng Chau. ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *arXiv:1704.01942 [cs, stat]*, August 2017.

John D Kelleher. *Deep learning*. MIT press, 2019.

Patrick Krüger and Hanno Gottschalk. Equivariant and steerable neural networks: A review with special emphasis on the symmetric group. *arXiv preprint arXiv:2301.03019*, 2023.

Pierre-Yves Lagrave and Frédéric Barbaresco. Introduction to Robust Machine Learning with Geometric Methods for Defense Applications. July 2021.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10.1109/5.726791.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, February 2015.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5058–5067, October 2017. doi: 10.1109/ICCV.2017.540.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. *arXiv:2010.09337 [cs, stat]*, October 2020.

Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally Occurring Equivariance in Neural Networks. *Distill*, 5(12):e00024.004, December 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.004.

Felipe Oviedo, Juan Lavista Ferres, Tonio Buonassisi, and Keith T. Butler. Interpretable and explainable machine learning for materials science and chemistry. *Accounts of Materials Research*, 3(6):597–607, jun 2022. doi: 10.1021/accountsmr.1c00244. URL `https://pubs.acs.org/doi/10.1021/accountsmr.1c00244`.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Markus Raffel, Christian E Willert, Fulvio Scarano, Christian J Kähler, Steve T Wereley, and Jürgen Kompenhans. *Particle image velocimetry: a practical guide*. springer, 2018.

Md Ashiqur Rahman and Raymond A Yeh. Truly scale-equivariant deep nets with fourier layers. *Advances in Neural Information Processing Systems*, 36, 2024.

Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to Compose Domain-Specific Transformations for Data Augmentation. *arXiv:1709.01643 [cs, stat]*, September 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. URL `https://arxiv.org/abs/1506.01497`.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Joseph J Rotman. *An introduction to the theory of groups*, volume 148. Springer Science & Business Media, 2012.

A Saranya and R Subhashini. A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends. *Decision analytics journal*, page 100230, 2023.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8429–8438, 2019. doi: 10.1109/ICCV.2019.00852.

Daniel Smilkov, Shan Carter, D. Sculley, Fernanda B. Viégas, and Martin Wattenberg. Direct-Manipulation Visualization of Deep Networks. *arXiv:1708.03788 [cs, stat]*, August 2017.

Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, December 2014. ISSN 0219-1377, 0219-3116. doi: 10.1007/s10115-013-0679-x.

Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A Bayesian Data Augmentation Approach for Learning Deep Models. *arXiv:1710.10564 [cs]*, October 2017.

Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

Maurice Weiler and Gabriele Cesa. General $E(2)$-Equivariant Steerable CNNs. *arXiv:1911.08251 [cs, eess]*, April 2021.

Thomas Wimmer, Vladimir Golkov, Hoai Nam Dang, Moritz Zaiss, Andreas Maier, and Daniel Cremers. Scale-equivariant deep learning for 3d data. *arXiv preprint arXiv:2304.05864*, 2023.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. doi: 10.1109/CVPR.2015.7298801.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. *arXiv:1506.06579 [cs]*, June 2015.

Xiaohang Zhang, Ling Wu, Zhengren Li, and Huayuan Liu. A robust method to measure the global feature importance of complex prediction models. *IEEE Access*, 9:7885–7893, 2021. doi: 10.1109/ACCESS.2021.3049412.

Appendix

A Related Work

A.1 Equivariant Neural Networks (ENNs)

Equivariant neural networks (ENN) [Krüger and Gottschalk, 2023] has become a popular research topic because models that are more equivariant have better generalization capability [Weiler and Cesa, 2021], an important goal of ML research. Equivariance sometimes occurs naturally [Olah et al., 2020], but guaranteeing equivariance requires more dedicated efforts. Various works focus on improving equivariance with respect to rotations [Weiler and Cesa, 2021, Assaad et al., 2022], shifts [Chaman and Dokmanić, 2021, Ding et al., 2023], and scales [Wimmer et al., 2023, Rahman and Yeh, 2024] through network architectural designs. Data augmentation during training is also effective for improving equivariance [Chen et al., 2020], with examples in generative models [Antoniou et al., 2018], Bayesian methods [Tran et al., 2017], and reinforcement learning [Ratner et al., 2017, Cubuk et al., 2019]. Existing work often implicitly assumes that more equivariant models will have lower errors when tested on large datasets, due to the close relationship between equivariance and robustness [Engstrom et al., 2019, Lagrave and Barbaresco, 2021]. While equivariance is indeed a close proxy for model robustness, the absence of evaluations directly showing it hinders more accurate understanding of model behaviors, which inspired our work on developing NERO evaluation.

A.2 Interpretable Machine Learning (IML)

Interpretable machine learning (IML) tackles the black-box nature of deep neural networks [Doshi-Velez and Kim, 2017] by employing various strategies focused on model components, model sensitivity, and surrogate models [Molnar et al., 2020]. Of the three, surrogate models [Ribeiro et al., 2018] is not described further here since they have little methodological connection to our work. Visualizations in IML seek to transform abstract data relationships into meaningful visual representations [Hohman et al., 2018]. Studies have shown that interactive visualization is a key aspect of sense-making when it comes to combining visual analytics with ML systems, which shapes our designs in presenting NERO evaluation through an interactive interface [Chatzimparmpas et al., 2020].

IML via Model Components. Existing IML works that focus on model components visualize the internals of a neural network. Abadi et al. [2016] developed the dataflow graphs in TensorFlow. Following this work, Smilkov et al. [2017] improved the dataflow graph by using visualization cues to represent weights sent between neurons. Beyond static visualizations, Yosinski et al. [2015] designed interactive visualizations of learned convolutional filters in neural networks, and Kahng et al. [2017] designed interactive system ActiVis for visualizing neural network responses to a subset of instances. While NERO evaluation does not visualize model components, it employs similar interactive visualization components.

IML via Feature Importance. Instead of visualizing model components, other approaches show feature importance by analyzing how model predictions change in response to changes in input data, in a way that is agnostic to the choice of ML model. Partial Dependent Plot (PDP) [Friedman, 2001] reveals the relationship between model predictions and one or two features by plotting the average change in model prediction when varying the feature value. Goldstein et al. [2014] built on this with Individual Conditional Expectation (ICE) plots that show model prediction changes due to changing features in individual data points, rather than the average. More recent works visualize expected conditional feature importance [Casalicchio et al., 2019], conduct sensitivity analysis [Štrumbelj and Kononenko, 2014], and further improve PDP with less computation cost [Apley and Zhu, 2019]. Lundberg and Lee [2017] present SHapley Additive exPlanations (SHAP) that assigns each feature an importance value to explain why a certain prediction is made. Zhang et al. [2021] derived a more robust, model-agnostic method from high-dimensional representations to measure global feature importance, which facilitates interpreting internal mechanisms of ML models. While NERO similarly employs data transformation and a response-recording mechanism, it does not visualize feature importance per se. Instead, it collects model responses with respect to data transformed by group actions as a whole, and supports visualizations at both aggregate (group) and instance levels.

B Mathematical Preliminaries

B.1 Group Action and Group Orbit

In this section, we give a concise summary of some elements of group theory, a rich topic meriting deeper consideration [Rotman, 2012]. A *group* G is a set with an operation "$\cdot$": $G \times G \to G$ that is associative ($(f \cdot g) \cdot h = f \cdot (g \cdot h)$), with an identity element e ($g \cdot e = e \cdot g = e$), and with inverses ($g \cdot g^{-1} = g^{-1} \cdot g = e$). A *group action* of group G on set X is a function $\phi : G \times X \to X$ that transforms an $x \in X$ by $g, h \in G$ according to $\phi(g, \phi(h, x)) = \phi(g \cdot h, x)$ and $\phi(e, x) = x$. The *orbit* of $x \in X$ under a group action ϕ is the set of all possible transformations $G(x) = \{\phi(g, x) | g \in G\}$. We use group orbits to generate a mathematically coherent family of ML model inputs, with which (human) users of the model can predict and reason about corresponding model outputs. For example, Fig. 13 illustrates a single 28×28 MNIST [Lecun et al., 1998] digit image x, and its orbit under the rotation group $SO(2)$ through the space X of all possible 28×28 images. We currently make NERO plots for spatial transformation group actions (shifts, rotations, flips), which have natural spatial layouts (e.g. the circular domain of $SO(2)$), but we want to highlight that NERO plots should in principle work with any group.

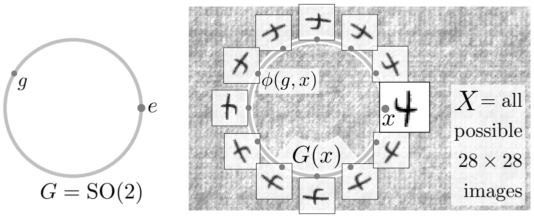


Figure 13: The group G of 2-D rotations, left, acts on the set X of images, right. An $x \in X$, a "4" digit, is rotated to $\phi(g, x)$ for a $g \in G$ via group action ϕ, part of the orbit $G(x) \subset X$ of all rotations of x.

B.2 Equivariance

Three terms – invariance, equivariance, covariance – for describing the relationship between changes in inputs and outputs of ML models [Marcos et al., 2017], can be introduced via a commutative diagram (2). The ML model hypothesis h maps from inputs X to outputs Y. For some group element g, actions $\phi(g)$ and $\tilde{\phi}(g)$ transform X and Y, respectively. Assuming (2) is true for some model (i.e., hypothesis h and transform $\tilde{\phi}(g)$ always reach the same output as input transform $\phi(g)$ followed by h), the following definitions describe *how*. The model is *invariant* with respect to the group action ϕ if $\tilde{\phi} = I$, the identity transform on Y. In classification tasks, invariance means that the classification result is unchanging while inputs are transformed in some way. A model is *equivariant* when the model inputs and outputs are transformed in the same way: $\phi = \tilde{\phi}$. For example, in object detection, where model outputs are object bounding boxes, if the object is shifted 5 pixels to the right, an equivariant model would predict the bounding box 5 pixels to the right. *Covariance* is an extension of equivariance in which ϕ and $\tilde{\phi}$ are mathematically distinct (because X and Y have distinct types), but have a semantic linkage necessitated by the structure of group G. For example, in optical flow, rotating the image inputs to a covariant model will produce an output in which both the vector field domain and the vectors themselves are correspondingly rotated. By a slight abuse of terminology, we use "equivariance" in this work to refer to all three commutative diagram properties.

$$\begin{tikzcd}
\text{model inputs} & X \arrow[r, "h"'] \arrow[d, "\phi(g)"'] & Y \arrow[d, "\tilde{\phi}(g)"] & \text{model outputs} \\
& X \arrow[r, "h"'] & Y &
\end{tikzcd} \tag{2}$$

C NERO Interface

While the specific components of the NERO interface are discussed and illustrated in §2, in this section we provide more design philosophy behind the actual interface. First of all, all the interfaces across different application domains as seen in §3 are designed with the general logic of overview on the left and details on the right. All sections are individually controllable and interactively linked. On the left, the dataset and subset of interest are selected via drop-down menus, with the resulting aggregate NERO plot below. The DR plot section supports choosing the scatterplot layout and coloring, and selection of individual data points within the scatterplot updates individual and detail views to the right. The individual NERO plot domain is the group orbit, and the interface permits moving within the orbit to look at a particular transform of a single sample, with real-time updates of the model output. In the MNIST interface, for example, clicking and dragging within the polar plot selects and changes the rotation angle, and updates the resulting rotated digit image and the models' outputs from it. Our interface is implemented in PySide (Python bindings for QT) as a desktop application, running on the same machine as the model.