Improving Equivariant Model Training via Constraint Relaxation

Stefanos Pertigkiozoglou[*]
University of Pennsylvania
pstefano@seas.upenn.edu

Evangelos Chatzipantazis[*]
University of Pennsylvania
vaghat@seas.upenn.edu

Shubhendu Trivedi
Independent
shubhendu@csail.mit.edu

Kostas Daniilidis
University of Pennsylvania
Archimedes, Athena RC
kostas@cis.upenn.edu

Abstract

Equivariant neural networks have been widely used in a variety of applications due to their ability to generalize well in tasks where the underlying data symmetries are known. Despite their successes, such networks can be difficult to optimize and require careful hyperparameter tuning to train successfully. In this work, we propose a novel framework for improving the optimization of such models by relaxing the hard equivariance constraint *during* training: We relax the equivariance constraint of the network's intermediate layers by introducing an additional non-equivariant term that we progressively constrain until we arrive at an equivariant solution. By controlling the magnitude of the activation of the additional relaxation term, we allow the model to optimize over a larger hypothesis space containing approximate equivariant networks and converge back to an equivariant solution at the end of training. We provide experimental results on different state-of-the-art network architectures, demonstrating how this training framework can result in equivariant models with improved generalization performance. Our code is available at https://github.com/StefanosPert/Equivariant_Optimization_CR

1 Introduction

The explicit incorporation of task-specific symmetry in the design and implementation of effective and parameter-efficient neural network (NN) models has matured into a rational and attractive NN design meta-formalism in recent years—that of group equivariant convolutional neural networks (GCNNs) (Cohen & Welling, 2016; Ravanbakhsh et al., 2017; Esteves et al., 2018; Kondor & Trivedi, 2018; Cohen et al., 2019; Maron et al., 2019; Weiler & Cesa, 2019; Bekkers, 2020; Villar et al., 2021; Xu et al., 2022; Pearce-Crump, 2023). GCNNs involve using the machinery of group and representation theory to compose layers that are equivariant to transformations of the input. Such networks, with hard-coded symmetry, have proven to be successful across a wide variety of tasks, while often affording significant data efficiency. Such tasks/domains include: RNA structure (Townshend et al., 2021), protein structure (Baek et al., 2021; Jumper et al., 2021), molecule generation (Satorras et al., 2021), medical imaging (Winkels & Cohen, 2019), natural language processing (Gordon et al., 2020; Petrache & Trivedi, 2024), computer vision (Chatzipantazis et al., 2023), robotics (Zhu et al., 2022; Ordoñez-Apraez et al., 2024), density functional theory (Gong et al., 2023), particle physics (Bogatskiy et al., 2020), lattice gauge theories (Boyda et al., 2021) amongst many others. GCNNs now have also matured enough to have a well-developed theory. This includes both prescriptive (or

[*]Equal Contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

architectural) theory and descriptive analysis. In general, GCNNs particularly stand out in domains with data scarcity, or with a high degree of symmetry (Kufel et al., 2023; Boyda et al., 2021), or in the physical sciences where respecting explicit symmetries could be dictated by physical laws, violating which could lead to physically implausible predictions.

Despite the successes of group equivariant models, there are several outstanding challenges that don't yet have general satisfactory solutions. We discuss two that have attracted recent attention. The first challenge—the primary motivation of our paper—has to do with the common observation that equivariant networks can be difficult to train (Wang et al., 2024; Kondor et al., 2018; Liao & Smidt, 2023). The reasons for this general difficulty are not well-understood, but it seems to occur in part because the training dynamics of such networks can be notably different from non-equivariant architectures. For instance, if a GCNN operates entirely in Fourier space (Bogatskiy et al., 2020; Kondor et al., 2018; Xu et al., 2022), most of the usual intuition about training NN models does not apply. Further, depending on the level of equivariance error tolerance for a task, the internal layers could be computationally intensive, and involve e.g. higher-order tensor products. Notably, the above difficulty arises even when the model is correctly specified i.e. the model and the data encode the same symmetry. The second challenge with GCNNs, has to do with the downsides of working with exact equivariance when the data itself might have some (possibly) relaxed symmetry. This has recently led to a spurt of work on developing more flexible networks that can vary the amount of equivariance depending on the task (Finzi et al., 2021; Romero & Lohit, 2022; van der Ouderaa et al., 2022; Wang et al., 2022; Huang et al., 2023; Petrache & Trivedi, 2023). Such models generally improve accuracy and will sometimes also simplify the optimization process as a side-effect. Broadly, proposed solutions involve adding additional regularization terms that penalize for relaxation errors, solving for the problem of model mis-specification (Petrache & Trivedi, 2023).

However, even though there is now work on relaxed[2] equivariant networks that addresses model mis-specification, existing works don't focus on improving the optimization process directly. In this paper, we take a step towards examining this question in more detail. We make the case that *even if we assume that the model is correctly specified*, relaxing the equivariance constraint during optimization and then projecting back to the equivariant space can itself help in improving performance. We conjecture that a prime reason for the optimization difficulty of GCNNs, as compared to non-equivariant models, is that their solution-space might be too severely constrained. We derive regularization terms that encourage each layer to operate in a larger hypothesis space during training—than being constrained to only be in the intertwiner space—while encouraging equivariant solutions. After the optimization is complete, we project the solution back onto the space of equivariant solutions. The approach can also be adapted to better optimize approximately equivariant networks in a similar manner. The focus of our work thus distinguishes it from works on relaxed equivariance—we are not concerned with mis-specification, but rather with isolating the optimization process itself.

Below we summarize the main contributions of our work:

- We present a novel training framework that can improve the performance of equivariant neural networks by relaxing the equivariance constraint during training and projecting back to the space of equivariant models during testing (as shown in Figure 1).
- We present a formulation that extends existing equivariant neural network architectures to be approximately equivariant. We show how training on the relaxed network can improve the performance of its equivariant subnetwork.
- We provide experimental evidence showcasing how our framework improves the performance of existing state-of-the-art equivariant architectures.

2 Related Work

There is little prior work on providing general procedures for improving the optimization process for equivariant neural networks directly. Elesedy & Zaidi (2021) sketched a projected gradient method to construct equivariant networks and suggested a regularization scheme that could be used to implement approximate equivariance. However, this was proposed as an aside in the paper (sections 7.2 and 7.3), without empirical or theoretical backing. Our work also involves a projected gradient procedure. However, the regularization scheme that we propose is substantially different.

[2]We use "relaxed" to encompass notions of partial and approximate equivariance (Petrache & Trivedi, 2023).

Work on approximate and partial equivariance (Finzi et al., 2021; Romero & Lohit, 2022; van der Ouderaa et al., 2022; Wang et al., 2022; Huang et al., 2023; Petrache & Trivedi, 2023; Wang et al., 2023) seems superficially related to ours, but comes with a different motivation. Such methods aim to match data symmetry with model symmetry and are not explicitly concerned with improving optimization. As a result, they are designed to address tasks with either inherent relaxed symmetries or tasks where the underlying relaxed symmetry is misspecified. Contrary to that, our method focuses on cases where the underlying symmetry is known exactly, and the relaxation of the equivariant constraint is used only during training as a way to improve the optimization. The works of Finzi et al. (2021); van der Ouderaa et al. (2022); Gruver et al. (2023); Otto et al. (2024); Petrache & Trivedi (2023) are nonetheless relevant since they provide methods for measuring relaxed equivariance, comprising of regularization schemes that are related to those used in our paper, since we also need measures of relaxation. In fact, the work of Gruver et al. (2023) directly inspires one component of our method. On the theoretical side, Petrache & Trivedi (2023) studied generalization-approximation tradeoffs in approximately/fully equivariant CNNs in a very general setting, characterizing the effect of model mis-specification on performance. They quantify equivariance as improving the generalization error, and the alignment of data and model symmetries as improving the approximation error. They leave the impact of improving the optimization error for future work. While we do not provide theoretical results, our work could be seen as focusing on optimization error component of the classical picture[3].

Maile et al. (2023) proposed what they call an *equivariance relaxation morphism*, which reparamterizes an equivariant layer to operate with equivariance constraints on a subgroup, but with the goal of architecture search. Flinth & Ohlsson (2023) provide an analysis of the optimization dynamics of equivariant models and compare them to non-equivariant models fed with augmented data. However, they don't use the analysis to provide insights on improving the optimization procedure itself.

Several researchers have recently tried to circumvent optimization difficulties in other ways. For instance, Mondal et al. (2023) suggests using equivariance-promoting canonicalization functions on top of large pre-trained models. The work of Basu et al. (2023b) operates with a similar motivation but without canonicalization. Yet another representative of work with a fine-tuning motivation, but in a different context is (Basu et al., 2023a). Finally, simplifying equivariant networks with heavy equivariant layers and improving their scalability is an active area of work and is related to easing optimization. Such works usually employ tools from representation theory, tensor algebra, or exploit sampling theorems over compact groups and their homogeneous spaces, such as Passaro & Zitnick (2023); Luo et al. (2024); Cobb et al. (2021); Ocampo et al. (2023).

3 Method

To introduce our proposed optimization framework, we first clearly define the equivariant constraint that the models we aim to train must satisfy. Assume a function $f : \mathbb{R}^n \to \mathbb{R}^m$ and a group G [4] acting on the input and output spaces via the general linear representations $\rho_{\text{in}} : G \to \text{GL}(\mathbb{R}^n)$, $\rho_{\text{out}} : G \to \text{GL}(\mathbb{R}^m)$. Then the function is said to be equivariant to the action of group G if for all $g \in G$ it satisfies the following constraint:

$$f(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)f(x), \quad \text{for all } x \in \mathbb{R}^n \tag{1}$$

Assuming we use a neural network to approximate the function above, the definition of equivariance as stated doesn't impose specific constraints on the individual layers of the network. Nevertheless, most of the current state-of-the-art equivariant architectures are a composition of simpler layers each one of which is constrained to be equivariant. In this case, the overall model is the result of a composition $f = f_N \circ f_{N-1} \circ \dots f_2 \circ f_1$ of simpler equivariant layers $f_i : V_i \to V_{i+1}$, where V_i, V_{i+1} are the input and output spaces on which the group G acts with the corresponding representations $\rho_{V_i}, \rho_{V_{i+1}}$ (assuming $V_1 = \mathbb{R}^n$, $V_N = \mathbb{R}^m$ are the input and output spaces respectively).

In this work we focus on a family of models as described above—that are defined through a composition of simpler equivariant linear layers. During standard training the linear layers are optimized over the set of intertwiners H_i, i.e. the set of linear maps between the representations (V_i, ρ_{V_i}) and $(V_{i+1}, \rho_{V_{i+1}})$ that have the equivariance property as stated in Equation 1. The set of intertwiners is only a subset of the set of all possible linear maps from V_i to V_{i+1}, and as a result, they have a reduced

[3] Characterizing model performance as generalization error + approximation error + optimization error

[4] We assume henceforth that we are always dealing with Matrix Lie groups.

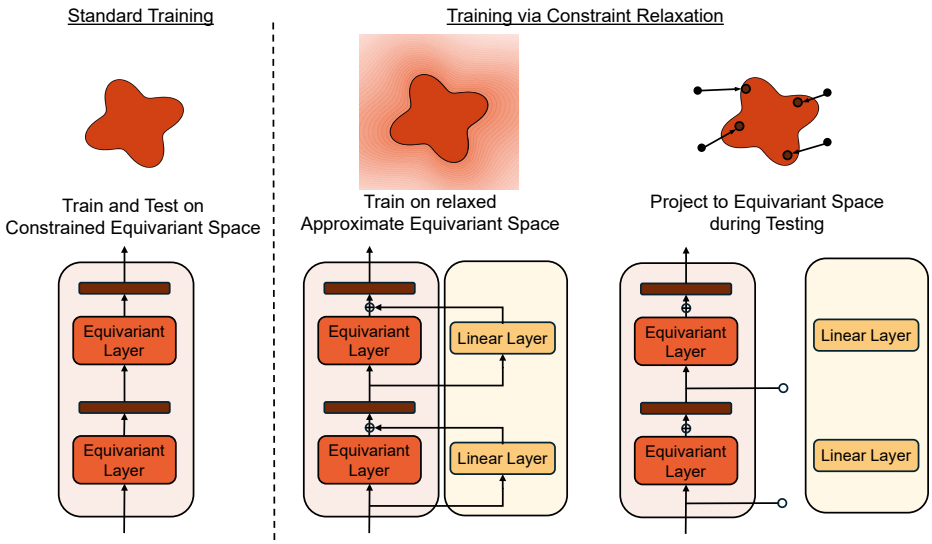


Figure 1: Standard training of equivariant NNs is constrained to a limited parameter space which can result in a challenging training process. We propose to relax these equivariant constraints during training, allowing optimization over a broader space of approximately equivariant models. During testing, we project the trained model back to the constrained space—arriving at an equivariant model with enhanced performance compared to equivalent models trained with the standard process.

number of free parameters. We propose to facilitate training over this constrained space by relaxing the constraint imposed on the intermediate layers and optimizing over a larger hypothesis space $\tilde{H}$ which is a superset of the set of equivariant models $H \subset \tilde{H}$. A trivial approach is to completely relax the constraint and solely optimize over the larger set containing all models. The problem with such an approach is that it completely abandons the concept of equivariance and all the attendant generalization benefits. Consequently, to expand the hypothesis space while keeping the benefits of equivariant models, we need a relaxation such that:

- Given a non-equivariant model $f \in \tilde{H}$, we can efficiently return to an equivariant one $\bar{f} \in H$.
- The relaxed model has a small *equivariance error* $P_{\text{ee}} = \text{E}_{x\sim p(x)} \int_G \|\rho_{out}(g)f(x) - f(\rho_{in}(g)x)\| dg$. This implies that although we extend the space of models we optimize over, we do not diverge too far away from the space of equivariant solutions.
- After we project back to the equivariant space, the error of the projection $P_{\text{pe}} = \text{E}_{x\sim p(x)} \left[\|f(x) - \bar{f}(x)\|\right]$ is also small. This ensures that while we optimize the less constrained model, we can return to the equivariant one without sacrificing the overall performance.

The first objective can be satisfied by defining an intermediate layer of the form:

$$f(x) = f_e(x) + \theta W x, \quad \theta \geq 0 \text{ and } f_e \in H, W \in \mathbb{R}^{|V_{\text{out}}| \times |V_{\text{in}}|} \tag{2}$$

where H is the set containing all possible equivariant solutions. Here it is easy to see that we can return to an equivariant model by setting $\theta = 0$, which we refer to as projection to the equivariant space. The formulation of the linear layer above is similar to the one used in the Residual Pathway Priors (RPP) (Finzi et al., 2021). Note that in RPP, the value of θ remains constant and acts as a prior on the level of equivariance we expect from a given task and dataset. Contrasted to that, in this work we aim to control the value of θ in order to actively change the level of equivariance during training and project back to the equivariance space during inference.

For the second objective, we need to utilize a metric that measures the relative distance of the model from the space of equivariant models H. It was observed by Gruver et al. (2023) that an easy way to measure how much a model satisfies the equivariant constraints is by using the norm of the Lie derivative. We present details in the next section.

3.1 Lie Derivative Regularization Term

Assume we are given a matrix Lie group G acting on a vector space V through its representation $\rho : G \to GL(V)$. For the given group there exists a corresponding Lie algebra $\mathfrak{g}$ with the property that for $A \in \mathfrak{g}$, $e^A \in G$. Additionally, there exists a corresponding Lie algebra representation $d\rho : \mathfrak{g} \to \mathfrak{gl}(V)$ such that $\rho(e^{tA}) = e^{d\rho(A)t}$.

If we take the derivative of the action of a group element $e^{tA} \in G$ at $t = 0$ we get the Lie derivative:

$$\left.\frac{d}{dt}\right|_{t=0} \rho(e^{tA}) = \left.\frac{d}{dt}\right|_{t=0} e^{d\rho(A)t} = d\rho(A) \tag{3}$$

Assume that the following group representation act on the vector space of functions as:

$$\rho_{\text{in}-\text{out}}(g)[f] = \rho_{\text{out}}(g)^{-1} \circ (f \circ \rho_{\text{in}}(g)) \tag{4}$$

As observed by Gruver et al. (2023) the lie derivative of the above action is zero for all equivariant functions f, since for all $g \in G$ the action $\rho_{\text{in}-\text{out}}(g)[f] = f$ is the identity map. As a consequence, we can use the norm of the Lie derivative as a metric to compute how much a function f deviates from the equivariant constraint of Equation 1. For the linear relaxation term Wx that we introduced in Equation 2, we have that the Lie derivative can be computed in a straightforward manner as:

$$\begin{aligned}\mathcal{L}_A(W) = \left.\frac{d}{dt}\right|_{t=0} \rho_{\text{out}}(e^{-At}) W \rho_{\text{in}}(e^{At}) &= \left.\frac{d}{dt}\right|_{t=0} e^{-d\rho_{\text{out}}(A)t} W e^{d\rho_{\text{in}}(A)t} \\ &= -d\rho_{\text{out}}(A)W + W d\rho_{\text{in}}(A)\end{aligned}$$

As a result, we can measure the degree that a linear layer satisfies the equivariant constraint at a point x, by computing the norm of the Lie derivative at that point for each one of the generators of the group. For example in the case where G is the group of 3D rotations ($G = \text{SO}(3)$), we can compute the Lie derivative for each generator:

$$J_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, J_y = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, J_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

During training, given an input distribution $p(x)$, we compute, for each linear layer Wx, the Lie derivative regularization term:

$$\mathcal{L}_{ld}(W) = \text{E}_{x \in p(x)} \left(\sum_{A \in \{J_i\}} \|\mathcal{L}_A(W)x\| \right)$$

Although the above regularization applies when the symmetry group we are considering is a matrix Lie Group, as we show in the experiments of Section 4.4, we can also define a similar regularizer for the case of discrete finite groups. In such a case, for a given linear layer with weights W and input x, we compute the sum of the norms of the difference $\mathcal{L}_{g_j}(W)x = (\rho(g_j)W - W\rho(g_j))\, x$ for all generators g_j of the discrete finite group under consideration.

As discussed in Otto et al. (2024) and shown in Figure 3(a), the inclusion of the above regularization terms encourages equivariant solutions and prevents the model from diverging away from the space of equivariant models. Moreover, Figure 2 shows how the inclusion of this regularization helps the overall training and results in a performance improvement of the final trained model.

3.2 Reducing the Projection Error

While we optimize over a larger hypothesis space, we always aim to return to an equivariant model after the end of training. Using the parametrization in Eq. 2 we can always do that by setting θ to be equal to zero. Although after the projection the resulting model is guaranteed to be equivariant, it might be far from the original relaxed version, meaning it might have a large projection error P_{pe}. Specifically, for an individual relaxed layer the projection error is:

$$\begin{aligned}P_{pe} = \text{E}_{x \sim p(x)} \left[\|f(x) - \bar{f}(x)\| \right] &= \text{E}_{x \sim p(x)} \left[\|f_e(x) + \theta W x - f_e(x)\| \right] \\ &= \text{E}_{x \sim p(x)} \left[\|\theta W x\| \right]\end{aligned}$$

As a result, to ensure that P_{pe} remains low we introduce a second regularization term on the norm $\|Wx\|$. Additionally, to reduce the contribution of θ on the projection error, we propose to schedule its value by slowly decreasing it during the last phase of training. Specifically, we apply a cyclic scheduling where given N_E total number of epochs, the value of θ at epoch i is:

$$\theta_i = \begin{cases} \frac{2i}{N_E} & \text{if } i < N_E/2 \\ 2 - \frac{2i}{N_E} & \text{if } i \geq N_E/2 \end{cases}$$

In Figure 2 we show how both the additional regularization term on the norm of the activation $\|Wx\|$, and the scheduling of θ, affect the performance of our framework.

3.3 Training Objective

Overall, given a task with a corresponding loss $\mathcal{L}_{\text{task}}$ and a data distribution D, our framework optimizes over the following training objective:

$$\mathcal{L} = \mathrm{E}_{(x,y)\sim D}\left[\mathcal{L}_{\text{task}}(f(x), y) + \lambda_{reg}\sum_{i=1}^{N}\left(\|W_i f_{i-1}(x)\| + \sum_{A\in J_i}\|\mathcal{L}_A(W_i) f_{i-1}(x)\|\right)\right] \tag{5}$$

where W_i is the weight matrix of the i^{th} additive unconstrained linear layer and $f_{i-1}(x)$ is the output of the $(i-1)^{\text{th}}$ layer (with f_0 corresponding to the input).

During training, we control the amplitude of the additive relaxation term by scheduling θ as described in Section 3.2. During inference, we evaluate only on the equivariant part of the model by setting $\theta = 0$. Thus as shown in Figure 1, after the end of training, the resulting model has the same parameter count and model architecture as the baseline model without any additional additive layers.

3.4 Relaxing the constraints of different equivariant architectures

In this section, we consider a selection of different equivariant architectures, and use them to illustrate how we could apply our proposed optimization framework:

Vector Neurons (Deng et al., 2021) In Vector Neurons, the primary linear layer processes features of the form $X \in \mathbb{R}^{N\times 3}$. It achieves equivariance by applying a left multiplication with a weight matrix $f(X) = WX$. This operation mixes only the rows of the input feature matrix and as a result when the input features rotate by R, the output also rotates since $f(XR) = WXR = f(X)R$.

To apply our proposed relaxation we add a linear layer that allows the mixing of all the elements of the input feature matrix. We can achieve this by unrolling the feature matrix into a vector of dimension (nm) and then after applying an unconstrainted linear layer, roll it back to a feature matrix. So the overall relaxed layer has the form:

$$f(X) = W_e X + \theta \text{uvec}\left[W\text{vec}(x)\right]$$

where vec, uvec are the corresponding unrolling and rolling operations.

SEGNN (Brandstetter et al., 2021) and Equiformer (Liao & Smidt, 2023): The intermediate representation of both SEGNN and Equiformer are steerable vector spaces that transform according to a representation of $\mathrm{SO}(3)$. In particular, both models process a collection of type l tensors that transform according to the Wigner-D matrices $D^{(l)}(g)$ of the corresponding type l. The interaction between tensors of different types can be done using the Clebsch-Gordan (CG) tensor product which is a bilinear operator that combines two input vectors x^{l_1}, x^{l_2} of types l_1 and l_2 and returns a tensor $(x^{l_1} \otimes x^{l_2})^l$ of type l as follows:

$$(x^{l_1} \otimes x^{l_2})^l_m = \sum_{m_1=-l_1}^{l_1}\sum_{m_2=-l_2}^{l_2} C^{(l,m)}_{(l_1,m_1)(l_2,m_2)} x^{(l1)}_{m_1} x^{(l_2)}_{m_2}$$

where $x^{(l1)}_{m_1}$, $x^{(l_2)}_{m_2}$ are the m_1^{th}, m_2^{th} elements of tensors $x^{(l1)}$, $x^{(l_2)}$ and $C^{(l,m)}_{(l_1,m_1)(l_2,m_2)}$ are the corresponding CG coefficients. In this operation, the CG coefficients restrict the possible interaction

between elements of different types of vectors. We relax the equivariant constraint by adding an unconstrained linear layer that can mix the elements from all the tensors used as intermediate representations, independent of their type. In Equiformer we add such a linear layer in the feed-forward network of the transformer block. Similarly in SEGNN, we add it to the layer that receives the aggregated messages from all the neighbors of a node and updates the node features.

Approximately Equivariant Steerable Convolutions (Wang et al., 2022): In this work, the authors designed approximate equivariant steerable convolutional layers. We apply our method by incorporating an additional unconstrained convolutional kernel. Since this task contains discrete symmetry groups, namely discrete rotations and scalings, we replace the Lie derivative regularizer with the corresponding one for discrete groups, described in Section 3.1.

4 Experiments

4.1 Equivariant Point Cloud Classification

We first evaluate our optimization framework by training different networks on the task of point cloud classification. We use the equivariant variants of PointNet (Qi et al., 2016) and DGCNN (Wang et al., 2019) which were proposed by Deng et al. (2021). We train on the ModelNet40 dataset (Chang et al., 2015), which contains 12311 point clouds from 40 different classes. We compare with the standard training of these networks using the same hyperparameter configuration as employed in Deng et al. (2021). During both training and testing, we sub-sample the input point clouds to 300 points.

To apply our method we relax the Vector Neurons linear layer by following the methodology described in Section 3.4. For both networks we set the regularization term $\lambda_{reg} = 0.01$, which is a value we use in all of the following experiments. We provide a more detailed description of the training parameters in Appendix A.1. Furthermore, in Appendix A.2 we describe the process of choosing the hyperparameter λ_{reg} and show the method's robustness to its value. Figure 2 showcases how applying our proposed framework benefits the training of both networks. Specifically, for the case of the smaller and less performant PointNet, we can see an even larger performance increase over the baseline. These results show how the performance benefits of our optimization framework increase in smaller under-parametrized networks, an effect we investigate further in Section 4.2. In Appendix A.3 we provide additional details on the computational and memory overhead of our proposed optimization, showcasing that while additional parameters are introduced during training the overhead in the training time is limited.

Ablations on the regularization terms and θ scheduling: In addition to the training curves of our method and of the baseline, Figure 2 shows the accuracy of our proposed optimization procedure when we remove some of the proposed regularization terms or the scheduling of θ. We observe that without any regularization both models diverge from the space of equivariant solutions. As a result, during inference when $\theta = 0$ their projection error P_{pe} becomes larger, resulting in a significant drop in test accuracy. Similarly, without the Lie derivative regularizer, the final test accuracy of both network variants drops. In such cases, Wx is unconstrained and can learn to extract non-equivariant features that the equivariant part f_e is not able to learn in any stage of the training. This effect can also be observed in figure 3(a) showing the total Lie derivative of the network when it is trained with and without the lie derivative regularization term. Not including the Lie derivative regularization allows the network, especially in the beginning of training, to optimize over solutions with large equivariance error. Finally, for both networks, we observe that θ scheduling, as described in Section 3.2, can benefit training compared to fixing θ to a constant low value. In Appendix A.4 we provide additional results showcasing how contrary to our method a model with a constant θ (without θ scheduling) has a significant drop in performance after it is projected into the equivariant space.

4.2 Scaling on Different Model and Dataset Sizes

To better understand how the model and dataset sizes affect our proposed optimization framework, we train models of variable depth on different numbers of training samples. As a baseline model we use the Steerable E(3) GNN (SEGNN) (Brandstetter et al., 2021) and we train it on the task of Nbody particle simulation (Kipf et al., 2018). This task consists of predicting the position of 5 electrically charged particles after 1000 time steps when given as input their initial positions, velocities, and charges.

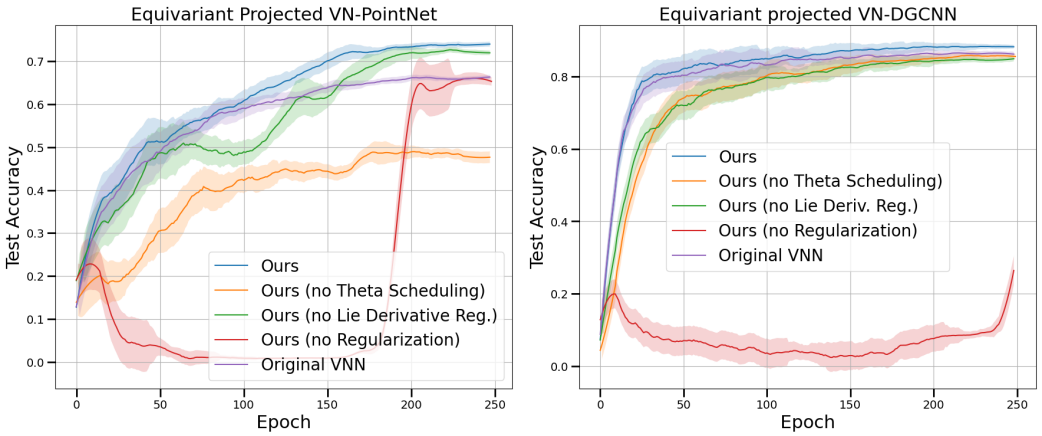
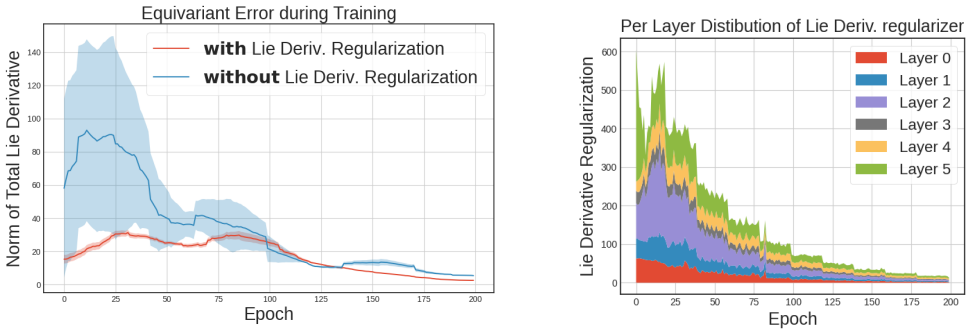


Figure 2: Test accuracy on ModelNet40 classification, during training of equivariant PointNet and DGCNN using a baseline training process and different versions of our method. The accuracy is computed for the equivariant models, i.e. for the models after they are projected in the equivariant space.



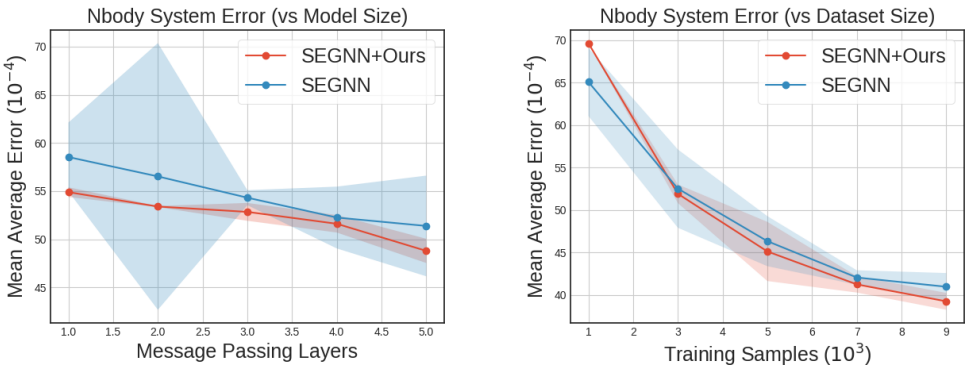
(a) (b)

Figure 3: (a) Norm of the total Lie derivative of the relaxed PointNet model trained with and without the Lie derivative regularization term. For the computation of the Lie derivative we use the method proposed in Gruver et al. (2023). (b) Value of the Lie derivative regularization term for each individual layer of the relaxed PointNet model while we train using our framework and with Lie derivative regularization weight set to $\lambda_{\text{reg}} = 0.01$

Figure 4(a) shows the mean average test error achieved by networks of different sizes, both when trained with a standard optimization, and when trained with our proposed framework. We can observe that for all sizes our method achieves better generalizations. The gap between our method and the baseline becomes greater in the cases of smaller networks, a phenomenon that we also observed in the point cloud classification experiments in Section 4.1. Thus, our framework, by relaxing the constraint and introducing additional degrees of freedom, can help the overall optimization, especially in models with a limited number of parameters. Additionally, figure 4(b) shows that when we fix the model size and increase the dataset size our method is able to scale better than the baseline. In both cases, we can observe that the training of the baseline has a much larger variance and is highly dependent on the random initialization of the layers. On the contrary, our method results in a more consistent training with a smaller variance between the random trials.

4.3 Molecular Dynamics Simulation

To evaluate our framework in a challenging task using a complex network architecture, we train Equiformer (Liao & Smidt, 2023) on the task of molecular dynamics simulations for a set of molecules provided as part of the MD17 dataset (Chmiela et al., 2017). The goal of this task is to predict the energy and forces from different configurations of a pre-specified molecule. Following Liao & Smidt (2023), for each molecule we use only 950 different configurations for training which significantly increases the task difficulty. For all training runs we use the same value of $\lambda_{reg} = 0.01$ as in the



(a) (b)

Figure 4: Mean Average Error on the Nbody particle simulation for (a) different model sizes, (b): different dataset sizes.

previous experiments and for the rest of the hyperparameters, we use the same configuration as the one proposed in Liao & Smidt (2023). In Table 1 we show that the mean absolute error of energy and force prediction achieved by Equiformer, both when it is trained using standard training, and when it is trained with our proposed optimization framework. Without any additional hyperparameter tuning, our framework is able to provide improvements on the performance of Equiformer even for this challenging data-scarce task.

Table 1: MAE of Equiformer trained with and without our optimization framework on a set of molecules from the MD17 dataset. The energy is reported in meV and the force in meV/Å units

	Aspirin		Benzene		Ethanol		Salicylic acid	
Methods	Energy	Forces	Energy	Forces	Energy	Forces	Energy	Forces
Equiformer	5.3	7.2	2.2	6.6	2.2	3.1	4.5	4.1
Equiformer+Ours	**5.2**	**7.1**	2.2	6.6	**2.0**	**2.9**	**4.1**	4.1

4.4 Optimizing Approximately Equivariant Networks

Finally, we show how our framework can be beneficial, not only for the optimization of exactly equivariant networks, but also for approximate equivariant ones. We apply our method on top of the approximate equivariant steerable CNNs proposed in Wang et al. (2022). Although these models are not projected back to the equivariant space, they are still regularized to stay within solutions with small equivariant error. The main difference with our framework is that in the approximate equivariant setting, the equivariant relaxation remains the same throughout training. On the contrary, we propose to progressively constrain the model by modulating the value of the unconstrained term by slowly decreasing the value of θ from Equation 2. As a result by applying our optimization framework on top of the standard training of the approximate equivariant kernels, we test how progressively introducing additional constraints throughout training can help the performance of the network.

We evaluate our method on the task of 2D smoke flow prediction described in Wang et al. (2022). The inputs of the model are sequences of successive 64×64 crops of a smoke simulation generated by PhiFlow (Holl et al., 2020). The desired output is a prediction of the velocity field for the next time step. We evaluate in two different settings: the "Future" setting where we evaluate on the same part of the simulation but we predict future time steps not included in the training, and the "Domain" setting where we evaluate on the same time steps as in training but in different spatial locations. The data are collected from simulations with different inflow positions and buoyant forces. For the rotational symmetry case, while the direction of the inflow and of the buoyant force is symmetric to 90° degrees rotations (C_4 symmetry group), the buoyancy factor changes for the different directions making the task not symmetric. For the scaling symmetry, the simulations are generated with different spatial and temporal steps, with the buoyant factor changing across scales.

Table 2: RMSE error on the synthetic smoke plume dataset with approximate rotational and scale symmetries. In the "Future" evaluation we train and evaluate the models in the same simulation location but we test for later time steps in the simulation from the ones used in training. In the "Domain" evaluation we train and evaluate the models on the same timesteps but on different spatial locations in the simulation.

Model		MLP	Conv	Equiv	Rpp	Lift	RSteer	RSteer+Ours
Rotation	Future	1.38 ± 0.06	1.21 ± 0.01	1.05 ± 0.06	0.96 ± 0.10	0.82 ± 0.08	0.80 ± 0.00	$\mathbf{0.79 \pm 0.01}$
	Domain	1.34 ± 0.03	1.10 ± 0.05	0.76 ± 0.02	0.83 ± 0.01	0.68 ± 0.09	0.67 ± 0.01	$\mathbf{0.58 \pm 0.00}$
Scale	Future	2.40 ± 0.02	0.83 ± 0.01	0.75 ± 0.03	0.81 ± 0.09	0.85 ± 0.01	0.70 ± 0.01	$\mathbf{0.62 \pm 0.02}$
	Domain	1.81 ± 0.18	0.95 ± 0.02	0.87 ± 0.02	0.86 ± 0.05	0.77 ± 0.02	0.73 ± 0.01	$\mathbf{0.67 \pm 0.01}$

In addition to the approximate equivariant steerable CNNs (RSteer), we compare with a simple MLP, with a non-equivariant convolutional network (ConvNet), as well as with an equivariant convolutional network (Equiv) (Weiler & Cesa, 2019) and with two additional approximate equivariant networks RPP (Finzi et al., 2021) and LIFT (Wang et al., 2021) that are trained using a standard training procedure. In Table 2 we see that by applying our optimization framework the resulting approximate equivariant model outperforms all other baselines in both cases of approximate rotational and scale symmetry. These results indicate that starting from an unconstrained model and progressively increasing the applied constraints can benefit optimization even in the case where at the end of training we stay in the space of approximate equivariant models and do not project back to the equivariant space.

5 Conclusion

In this work, we focus on the optimization of equivariant NNs. We proposed a framework for improving the overall optimization of such networks by relaxing the equivariance constraint and optimizing over a larger space of approximately equivariant models. We showcase the importance of utilizing regularization during training to ensure that the relaxed models stay close to the space of equivariant solutions. After training, we project back to the equivariant space arriving at a model that respects the data symmetries, while retaining its high performance on the task. We evaluate our proposed framework and its individual components over a variety of different equivariant network architectures and training tasks, and we report that it can consistently provide performance benefits over the standard training procedure. A theoretical analysis of our approach, possibly with appeal to empirical process theory (Pollard, 1990) to control the optimization error, is left for future work.

Acknowledgements

We gratefully acknowledge support by the following grants: NSF FRR 2220868, NSF IIS-RI 2212433, ARO MURI W911NF-20-1-0080, and ONR N00014-22-1-2677.

References

Baek, M., Dimaio, F., Anishchenko, I. V., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A. M., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., Opperman, D. J., Sagmeister, T., Buhlheller, C., Pavkov-Keller, T., Rathinaswamy, M. K., Dalwadi, U., Yip, C. K., Burke, J. E., Garcia, K. C., Grishin, N. V., Adams, P. D., Read, R. J., and Baker, D. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373: 871 – 876, 2021.

Basu, S., Katdare, P., Sattigeri, P., Chenthamarakshan, V., Campbell, K. D., Das, P., and Varshney, L. R. Efficient equivariant transfer learning from pretrained models. In *Neural Information Processing Systems*, 2023a. URL `https://api.semanticscholar.org/CorpusID:258740850`.

Basu, S., Sattigeri, P., Ramamurthy, K. N., Chenthamarakshan, V., Varshney, K. R., Varshney, L. R., and Das, P. Equi-tuning: Group equivariant fine-tuning of pretrained models. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth*

Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pp. 6788–6796. AAAI Press, 2023b. doi: 10.1609/AAAI.V37I6.25832. URL https://doi.org/10.1609/aaai.v37i6.25832.

Bekkers, E. J. B-spline cnns on lie groups. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=H1gBhkBFDH.

Bogatskiy, A., Anderson, B., Offermann, J., Roussi, M., Miller, D., and Kondor, R. Lorentz group equivariant neural network for particle physics. In *International Conference on Machine Learning*, pp. 992–1002. PMLR, 2020.

Boyda, D., Kanwar, G., Racanière, S., Rezende, D. J., Albergo, M. S., Cranmer, K., Hackett, D. C., and Shanahan, P. E. Sampling using su (n) gauge equivariant flows. *Physical Review D*, 103(7): 074504, 2021.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E., and Welling, M. Geometric and physical quantities improve e(3) equivariant message passing. 2021.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Chatzipantazis, E., Pertigkiozoglou, S., Dobriban, E., and Daniilidis, K. $\mathrm{SE}(3)$-equivariant attention networks for shape reconstruction in function space. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=RDy3IbvjMqT.

Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., and Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017. doi: 10.1126/sciadv.1603015. URL https://www.science.org/doi/abs/10.1126/sciadv.1603015.

Cobb, O., Wallis, C. G. R., Mavor-Parker, A. N., Marignier, A., Price, M. A., d'Avezac, M., and McEwen, J. Efficient generalized spherical {cnn}s. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rWZz3sJfCkm.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *ICML*, 2016.

Cohen, T. S., Geiger, M., and Weiler, M. A general theory of equivariant CNNs on homogeneous spaces. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. Vector neurons: a general framework for so(3)-equivariant networks. *arXiv preprint arXiv:2104.12229*, 2021.

Elesedy, B. and Zaidi, S. Provably strict generalisation benefit for equivariant models. In *International conference on machine learning*, pp. 2959–2969. PMLR, 2021.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 52–68, 2018.

Finzi, M., Benton, G., and Wilson, A. G. Residual pathway priors for soft equivariance constraints. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Flinth, A. and Ohlsson, F. Optimization dynamics of equivariant and augmented neural networks, 2023.

Gong, X., Li, H., Zou, N., Xu, R., Duan, W., and Xu, Y. General framework for e (3)-equivariant neural network representation of density functional theory hamiltonian. *Nature Communications*, 14(1):2848, 2023.

Gordon, J., Lopez-Paz, D., Baroni, M., and Bouchacourt, D. Permutation equivariant models for compositional generalization in language. In *ICLR*, 2020.

Gruver, N., Finzi, M. A., Goldblum, M., and Wilson, A. G. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JL7Va5Vy15J.

Holl, P. M., Um, K., and Thuerey, N. phiflow: A differentiable pde solving framework for deep learning via physical simulations. In *Workshop on Differentiable Vision, Graphics, and Physics in Machine Learning at NeurIPS*, 2020.

Huang, N., Levie, R., and Villar, S. Approximately equivariant graph networks. *ArXiv*, abs/2308.10436, 2023.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2688–2697. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kipf18a.html.

Kondor, R. and Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, 2018.

Kondor, R., Lin, Z., and Trivedi, S. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31, 2018.

Kufel, D., Kemp, J., and Yao, N. Y. Approximately-invariant neural networks for quantum many-body physics. 2023. URL https://api.semanticscholar.org/CorpusID:268031561.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Luo, S., Chen, T., and Krishnapriyan, A. S. Enabling efficient equivariant operations in the fourier basis via gaunt tensor products. *arXiv preprint arXiv:2401.10216*, 2024.

Maile, K., Wilson, D. G., and Forré, P. Equivariance-aware architectural optimization of neural networks, 2023.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. *ArXiv*, abs/1812.09902, 2019.

Mondal, A. K., Panigrahi, S. S., Kaba, S.-O., Rajeswar, S., and Ravanbakhsh, S. Equivariant adaptation of large pretrained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=m6dRQJw280.

Ocampo, J., Price, M. A., and McEwen, J. Scalable and equivariant spherical CNNs by discrete-continuous (DISCO) convolutions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=eb_cpjZZ3GH.

Ordoñez-Apraez, D., Turrisi, G., Kostic, V., Martin, M., Agudo, A., Moreno-Noguer, F., Pontil, M., Semini, C., and Mastalli, C. Morphological symmetries in robotics. *arXiv preprint arXiv:2402.15552*, 2024.

Otto, S. E., Zolman, N., Kutz, J. N., and Brunton, S. L. A unified framework to enforce, discover, and promote symmetry in machine learning, 2024. URL https://arxiv.org/abs/2311.00212.

Passaro, S. and Zitnick, C. L. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In *International Conference on Machine Learning*, pp. 27420–27438. PMLR, 2023.

Pearce-Crump, E. Brauers group equivariant neural networks. In *International Conference on Machine Learning*, pp. 27461–27482. PMLR, 2023.

Petrache, M. and Trivedi, S. Approximation-generalization trade-offs under (approximate) group equivariance. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 61936–61959. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c35f8e2fc6d81f195009a1d2ae5f6ae9-Paper-Conference.pdf.

Petrache, M. and Trivedi, S. Position paper: Generalized grammar rules and structure-based generalization beyond classical equivariance for lexical tasks and transduction. *arXiv preprint arXiv:2402.01629*, 2024.

Pollard, D. Empirical processes: Theory and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, pp. i–86. JSTOR, 1990.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2016. URL http://arxiv.org/abs/1612.00593. cite arxiv:1612.00593.

Ravanbakhsh, S., Schneider, J. G., and Póczos, B. Equivariance through parameter-sharing. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2892–2901. PMLR, 2017. URL http://proceedings.mlr.press/v70/ravanbakhsh17a.html.

Romero, D. W. and Lohit, S. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022.

Satorras, V. G., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E(n) equivariant normalizing flows for molecule generation in 3d. *ArXiv*, abs/2105.09016, 2021.

Townshend, R. J. L., Eismann, S., Watkins, A. M., Rangan, R., Karelina, M., Das, R., and Dror, R. O. Geometric deep learning of rna structure. *Science*, 373:1047 – 1051, 2021.

van der Ouderaa, T. F. A., Romero, D. W., and van der Wilk, M. Relaxing equivariance constraints with non-stationary continuous filters. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, Dec 2022. doi: 10.48550/ARXIV.2204.07178. URL https://arxiv.org/abs/2204.07178.

Villar, S., Hogg, D. W., Storey-Fisher, K., Yao, W., and Blum-Smith, B. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in Neural Information Processing Systems*, 34:28848–28863, 2021.

Wang, D., Walters, R., Zhu, X., and Platt, R. Equivariant q learning in spatial action spaces. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=IScz42A3iCI.

Wang, R., Walters, R., and Yu, R. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*. PMLR, 2022.

Wang, R., Walters, R., and Smidt, T. Relaxed octahedral group convolution for learning symmetry breaking in 3d physical systems. In *NeurIPS 2023 AI for Science Workshop*, 2023. URL https://openreview.net/forum?id=B8EpSHEp9j.

Wang, R., Hofgard, E., Gao, H., Walters, R., and Smidt, T. E. Discovering symmetry breaking in physical systems with relaxed group convolution, 2024. URL https://arxiv.org/abs/2310.02299.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

Weiler, M. and Cesa, G. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Winkels, M. and Cohen, T. Pulmonary nodule detection in ct scans with equivariant cnns. *Medical image analysis*, 55:15–26, 2019.

Xu, Y., Lei, J., Dobriban, E., and Daniilidis, K. Unified fourier-based kernel and nonlinearity design for equivariant networks on homogeneous spaces. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24596–24614. PMLR, 2022. URL https://proceedings.mlr.press/v162/xu22e.html.

Zhu, X., Wang, D., Biza, O., Su, G., Walters, R., and Platt, R. Sample efficient grasp learning using equivariant models. *CoRR*, abs/2202.09468, 2022. URL https://arxiv.org/abs/2202.09468.

A Appendix/ Supplemental Material

A.1 Training Details

In this section, we provide additional details for the application of our framework in the experiments presented in this work. We fix the weight of the regularization term to be $\lambda_{reg} = 0.01$ for all the experiments. We arrive that the above value for the hyperparameter λ_{reg} by performing grid-search using cross validation, as described in more detail in Section A.2. Additionally, except on the corresponding ablation study, we use the scheduling of the value of θ as described in Section 3.2. The variance reported is over 5 random trials of the same experiment with different seeds. We run all the experiments on NVIDIA A40 GPUs. For the model-specific training details:

- **Point Cloud Classification**: We use as baselines the VN-PointNet and VN-DGCNN network architectures described in the work of Deng et al. (2021). For the relaxed version of VN-PoitNet we train for 250 epochs using the Adam optimizer (Kingma & Ba, 2015), with an initial learning rate of 10^{-3}, that we decrease every 20 epochs by a factor of 0.7, and weight decay equal to 10^{-4}. For the relaxed version of VN-DGCNN we train for 250 epochs using stochastic gradient descent, with an initial learning rate of 10^{-3}, that we decrease using cosine annealing, and weight decay equal to 10^{-4}. The batch size used was 32.
- **Nbody particle simulation**: We train the relaxed version of SEGNN (Brandstetter et al., 2021) for 1000 epochs using Adam optimizer (Kingma & Ba, 2015) with a learning rate of $5 * 10^{-4}$, a weight decay of 10^{-12} and batch size of 100. We report the test MAE for the model at the training epoch that achieved the minimum validation error.
- **Molecular Dynamics Simulation**: We train the relaxed version of Equiformer (Liao & Smidt, 2023) for 1500 epochs using AdamW optimizer (Loshchilov & Hutter, 2019) with an initial learning rate of 10^{-5}, that we decrease using cosine annealing, and with weight decay equal to 10^{-6}. The batch size used was 8. We use the network variant with max representation type set to $L_{\max} = 2$
- **Approximately Equivariant Steerable Convolution**: We train the approximately equivariant steerable convolution proposed in Wang et al. (2022) after we apply our additional relaxation. We modify the same architecture used in the original work which contains 5 layers of approximate equivariant steerable convolutions. We train for 1000 epochs using the Adam optimizer (Kingma & Ba, 2015). We use an initial learning rate of 10^{-4}, that we decrease at each epoch by 0.95. We perform early stopping, where the stopping criterion is that the mean validation score of the last 5 epochs exceeds the mean validation score of the previous 5 epochs.

A.2 Choice of Hyperparameters

In all the experiments, apart from the weight λ_{reg} of the proposed regularization term, we use the same hyperparameters as the ones used by the baseline methods we compare with. For the choice of λ_{reg} we perform hyperparameter grid search using cross-validation with an 80%-20% split of the original training set of ModelNet40 into training and validation. Figure 5 showcases the performance of a VN-Pointnet model trained with our method on the 80% training split and evaluated on the 20% validation split for different values of λ_{reg}. We observed that the best value of λ_{reg} is relatively robust across tasks, so we performed an extensive hyperparameter search for the task of point cloud classification, and we used the found value $\lambda_{reg} = 0.01$ across all other tasks.

A.3 Computational and Memory Overhead of proposed method

The computational overhead introduced by our method only affects the training process. During inference, after the projection to the equivariant space, the retrieved model has the same architecture as the corresponding base model to which we apply our method on, thus it also has the same computational and memory requirements. In Table 3 we show the cost of our method, in terms of training time and number of additional parameters. While our proposed method introduces additional parameters during training, due to the parallel nature of the unconstrained non-equivariant term, the overhead in training time can be limited given enough memory resources. As a result, while the additional parameters are approximately three times the parameters of the base model the increase in the training time is below 10% of the base training time.

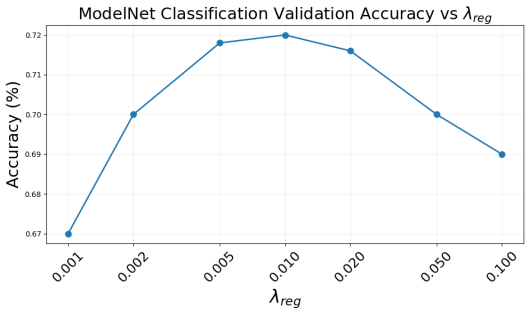


Figure 5: ModelNet40 classification accuracy on the validation set using our proposed method with different values of λ_{reg}. The base model used was the VN-PointNet. The model was trained on a split of the training set containing 80% of the training data. The other 20% of the data were held out as the validation set used to evaluate the model.

Table 3: Additional Number of parameters and Computational Overhead introduced by the proposed method

Model	**Number of Parameters (Base Model)**	**Additional Parameters (Ours)**	**Time per Epoch (Base Model)**	**Time per Epoch (Ours)**
VN-PointNet	1.9M	6.4M	75s	80s
VN-DGCNN	1.8M	6.2M	148s	154s
Equiformer	3.4M	10M	52s	57s

A.4 Ablation on θ Scheduling and Equivariant projection

During the later stages of training our proposed θ scheduling slowly decreases the level of relaxation of the equivariant constraint, bringing the model closer to the equivariant space. This process allows the model to smoothly transition from the relaxed equivariant case to the exact equivariant one. In Figure 6 we show a comparison of the performance of a model trained with our proposed θ scheduling and a model trained with a constant θ before and after it is projected to the equivariant space. While the performance of the relaxed equivariant model with constant θ is close to the performance achieved by our method, we can observe a sudden drop in performance once it is projected back to the equivariant space. On the other hand, our proposed scheduling of θ allows the model to return to the equivariant space by the end of training without showcasing such a significant performance drop.

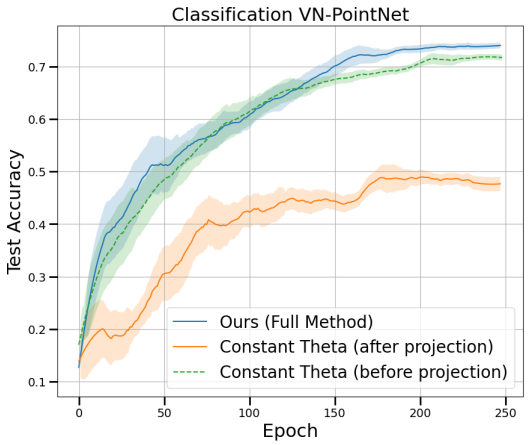


Figure 6: Comparison of the performance on ModelNet40 classification (300 points), for a model trained with our method and a model trained with a constant value of θ, for which the level of equivariant error is controlled only by the regularization term. For the latter method we show results both before and after the projection to the equivariant space.

A.5 Additional comparison with Methods using Equivariant Adaptation/Fine Tuning

As described in Section 2, while our work focuses on improving the optimization of equivariant networks, the works of Mondal et al. (2023) (Equiv-Adapt) and Basu et al. (2023b) (Equi-Tuning) focus on circumventing the need of optimizing equivariant network by performing equivariant adaptation or fine tunining of non-equivariant models. Since such methods have a different focus, mainly utilizing already pre-trained non-equivariant models to solve equivariant tasks, a straightforward comparison can be challenging. Nevertheless, in Table 4 we provide a comparison with our proposed method on the task of sparse point cloud classification. We can see that our method outperforms Equiv-Adapt and has performance close to the one achieved by Equi-Tuning which requires multiple forward passes during inference.

Table 4: Comparison of our proposed method with previous works performing equivariant adaption or finetuning, on ModelNet40 classification (Base model: VN-PointNet). Here it is important to note that in the case of Equi-Tuning, equivariance is achieved by group averaging. As a result, during inference the model is required to perform multiple forward passes, which slows down the method's inference.

Equiv-Adapt	Equi-Tuning	Original VNN	$\boldsymbol{Ours}$
66.3%	74.9%	66.4%	74.5%

A.6 Limitations

As we describe in Section 3, our work focuses on the assumption that the equivariant NN satisfies the equivariant constraint by imposing it in all of its intermediate layers. Although this assumption is general enough to cover a large range of state-of-the-art equivariant architecture, it doesn't apply to all possible equivariant networks since it is possible to ensure overall constraint satisfaction using a different methodology. Additionally the proposed regularizers in Section 3.1 can be applied to tasks where the symmetry group is a matrix Lie group or a discrete finite group. Extending our proposed framework to arbitrary symmetry groups is a future research question that is not addressed in this paper.

A.7 Broader Impact

This paper focuses on the question of improving the optimization of equivariant neural networks. Such equivariant networks are currently used to solve tasks in different fields– ranging from computer vision to computational chemistry. As a result, its broader societal impact is highly dependent on the specific network it enables optimizing.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims and contributions of the paper are included at the end of the introduction in Section 1. Similarly, the abstract of the paper reflects these claims

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Appendix A.6

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper doesn't provide any theoretical result in the form of a proof

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In the experimental Section 4 and in the supplementary material in Section A.1, we disclose all the information needed to reproduce our results by citing the appropriate previous works. Additionally, we provide the set of hyperparameters and training details that are novel for our work. Finally in Section 3.4 we provide a description of how our framework can be implemented for the set of network architectures that we use in the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In the abstract, we provide a link to a GitHub repository containing all the code and instructions necessary to implement our proposed optimization method. Additionally, all of the data used in the experiments are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the required configurations for the different experiments are provided in the Section 4 and Section A.1 of the appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the variance of our experimental results to random initialization for all the experiments except the molecular dynamic simulation 4.3 due to its high computational cost.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section A.1 of the appendix we disclose the computational resources used for the experiments in this work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact of our work in Section A.7

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not plan to release any data or pre-trained models

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper uses only publicly available data and code for which it credits their creators appropriately.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper doesn't release any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper doesn't include crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper doesn't include crowdsourcing or any research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.