
Final Report:

Equivariant Adaptation of Large Pretrained Models

Author: Giacomo Negri — Matrikelnummer: 03815415 — Date: January 31, 2026

Abstract

Standard large-scale pretrained models typically lack equivariance to geometric transformation such as rotations, which limits their performance when subject to input distribution shift. Group equivariant neural network can address such symmetry by design, but they requires specialized layers and costly training. This report reviews Equivariant Adaptation of Large Pretrained Models (1), a wrapper based on learned canonicalization (2), where inputs are transformed to a canonical pose before being processed by a frozen or fine-tuned pretrained backbone. A key contribution is the use of a canonicalization prior, based on the dataset, that reduces misalignment with the backbone’s training distributon, aiming at preserving in-distribution (ID) performance while enhancing transformation robustness.

1. Introduction

Large pretrained models, such as Vision Transformers and PointNet, achieve state of the art performance by being trained on massive datasets. However, typically they are highly sensible to out-of-distribution (OOD) inputs. Therefore, achieving equivariance, meaning a predictable transformation of the output given a transformation of the input, usually involves redesigning the internal layers (architectural equivariance), which is often computationally expensive for large models.

The research goal of the paper ”Equivariant Adaptation of Large Pretrained Models” is to develop a method that provides equivariance as a ”drop-in” wrapper (EquiAdapt) for any pretrained backbone without the need to modify its internal weights or architecture.

2. Methodology

The idea behind this paper comes from the *single-view-plus-transformation* model for invariant shape recognition, where it is suppose that humans convert an object to a canonical orientation by a transformation process before recognize

it (3). Similarly, the authors aim to learns how to rotate the input data back to a canonical orientation before the backbone sees it and returns its prediction.

2.1. Groups and equivariance

A group $\mathcal{G}$ is a set equipped with associative binary operation such that is closed under this operation and each element $g \in \mathcal{G}$ has a unique inverse. A group representation is a mapping from a group $\mathcal{G}$ to an invertible linear operators: $\rho : \mathcal{G} \rightarrow \text{GL}(\mathcal{X})$, where a linear action is: $\rho(g) \cdot \mathbf{x}$, for each $\mathbf{x} \in \mathcal{X}$. Therefore, we can say that a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is equivariant if:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(g) \cdot f(\mathbf{x}) \quad (1)$$

particularly, if $\rho'(g) = \mathbf{I}, \forall g \in \mathcal{G}$ then f is invariant.

2.2. Canonicalization framework

A general way to enforce equivariance is to use orbit induced by the group $\mathcal{G}$. For any $\mathbf{x} \in \mathcal{X}$, the orbit $\mathcal{O}_{\mathbf{x}} = \{\rho(g) \cdot \mathbf{x} : g \in \mathcal{G}\}$ represents all the transformed version of the specific sample. Let’s consider now $p : \mathcal{X} \rightarrow \mathcal{Y}$ a pretrained model, $c : \mathcal{X} \rightarrow \mathcal{G}$ the canonicalization function for the group $\mathcal{G}$, and $\rho : \mathcal{G} \rightarrow \text{GL}(\mathcal{X})$ the group representation function. We can define a function $f(\mathbf{x})$ as:

$$f(\mathbf{x}) = \rho'(c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x}))^{-1} \cdot \mathbf{x}) \quad (2)$$

where we are first canonicalizing, $\rho(c(\mathbf{x}))^{-1} \cdot \mathbf{x}$, then predicting, $p(\cdot)$, and finally restoring such prediction, $\rho'(c(\mathbf{x}))$, as illustrated in Fig. 1.

Crucially, a sufficient condition for eq. 2 to be equivariant is if c is $\mathcal{G}$ -equivariant:

$$c(\rho(g) \cdot \mathbf{x}) = \rho(g) \cdot c(\mathbf{x}) \quad \forall g \in \mathcal{G}, \mathbf{x} \in \mathcal{X} \quad (3)$$

Effectively decoupling equivariance from the prediction task through Learned Canonicalization (LC), as shown by (2).

Frequently, the prediction network p is already equivariant to a subgroup $\mathcal{K} \leq \mathcal{G}$ (e.g. translations for CNN). As shown in (2), the canonicalization framework supports partial canonicalization, where the canonicalizer is responsible only for the missing symmetries.

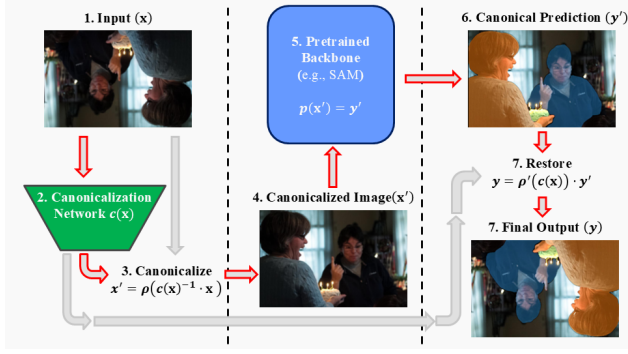


Figure 1. Representative example of how equivariance via canonicalization is achieved for a segmentation task.

An important consideration to be done is the following: if $\mathbf{x}$ has non-trivial stabilizer $\mathcal{G}_{\mathbf{x}} = \{g \in \mathcal{G} : \rho(g) \cdot \mathbf{x} = \mathbf{x}\}$ it might be impossible to select a unique canonical representation, because multiple elements of the stabilizer correspond to the same observed sample. To address this, (2) introduce *relaxed equivariance*, where equivariance is required only up to a stabilizer coset:

$$\forall g_1, \mathbf{x} \quad \exists g_2 \in g_1 \mathcal{G}_{\mathbf{x}} : \quad h(\rho(g_1) \cdot \mathbf{x}) = \rho'(g_2) \cdot h(\mathbf{x}) \quad (4)$$

which allows the canonicalizer to produce asymmetric outputs from symmetric inputs. Such symmetry breaking is essential in physics and chemistry where symmetric starting state can lead to asymmetric results. We often refer to this phenomena as Curie’s principle.

2.3. Dataset’s prior and regularization

The paper’s innovation stems from the introduction of a *canonicalization prior* to resolve two problems with a “naive” application of canonicalization:

1. **Alignment problem:** canonicalization may turn data into orientations that are mathematically valid but misaligned with the training distribution of the backbone, causing poor performance (1).
2. **Augmentation problem:** at the beginning of training, randomly initialized canonicalizer may cause excessive data augmentation. For those datasets that require only small transformations, this variance can lead to unstable training and lower performance on ID samples (1).

Therefore, to ensure canonical orientations remain consistent with the pretrained model’s preferences, the authors introduce a prior loss based on the Kullback-Leibler (KL) divergence:

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [D_{KL}(\mathbb{P}_{\mathcal{D}} \| \mathbb{P}_{c(\mathbf{x})})] \quad (5)$$

where $\mathbb{P}_{\mathcal{D}}$ represents the prior distribution over the group $\mathcal{G}$ (typically a distribution $q(R)$ centered at the identity), and $\mathbb{P}_{c(\mathbf{x})}$ is the one predicted by the canonicalization network.

Expanding the KL divergence for a fixed prior $q(R)$ and a predicted density $p(R|c(\mathbf{x}))$ we get:

$$D_{KL}(q \| p) = \int_{\mathcal{G}} q(R) \log q(R) dR - \int_{\mathcal{G}} q(R) \log p(R|c(\mathbf{x})) dR$$

Because the prior distribution $q(R)$ is constant, the first term is a constant C during optimization. By consequence, minimizing the KL divergence is mathematically equivalent to minimizing the cross-entropy, leading to an actual implementation through negative log-likelihood:

$$\mathcal{L}_{\text{prior}} = -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{R \sim q} [\log p(R|c(\mathbf{x}))] \quad (6)$$

An instance of such process is represented in Fig. 2

2.4. Discrete and continuous rotations

For discrete rotations (e.g. the cyclic group C_n), the prior is defined as a Dirac delta $\delta_{R, I}$ at the identity, simplifying the loss to $-\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \log p_{c(\mathbf{x})}(\mathbf{I})$. In the case of continuous rotations, the objective is defined by using the isotropic version of the matrix Fisher distribution on $SO(n)$:

$$p(R|\hat{R}, s) = \frac{1}{n(s)} \exp \left(\text{Tr}(\hat{R}^T R) \right) \quad (7)$$

where $R \in SO(n)$ is a random rotation matrix, while $\hat{R}$ and s are, respectively, the mode and the concentration around different axis of the distribution. Specifically, by Proposition 1 of (1) it can be shown that given a prediction $p(R|\hat{R}_p, s_p)$ and a prior $q(R|\hat{R}_q, s_q)$, where $\hat{R}_q = \mathbf{I}$, the canonicalization prior loss is:

$$\mathcal{L}_{\text{prior}} = -\lambda \text{Tr}(\mathbf{R}_{c(\mathbf{x})}) = \frac{\lambda}{2} \|\mathbf{R}_{c(\mathbf{x})} - \mathbf{I}\|_F \quad (8)$$

where $\lambda = N(s_q)s_p$.

2.5. Technical implementation and optimization

From a practical perspective the representations were achieved as follows.

Discrete the canonicalization network, G-CNN (4) or E(2)-Steerable Network (5), are designed to outputs logits corresponding to each group element along a fibre dimension. Logits are then transformed through softmax, by using the straight-through gradient (6) estimator to deal with the non-differentiability of argmax during backpropagation.

Continuous for image domain two vector fields are outputted by E(2)-Steerable Network, then spatially averaged and Gram Schmidt orthonormalized. For point clouds, they

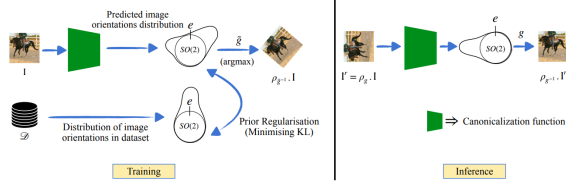


Figure 2. Training and inference with regularized canonicalization method. The canonicalization function outputs a distribution over image orientations used to canonicalize the input image. Additionally, during training, this predicted distribution is regularized to match the orientations seen in the dataset. (Source (1))

used the same set up as in (2) where they create a $E(d)$ -equivariant Network by defying the canonicalization function as $h(x) = \rho(h^O(x), h^t(x))$, where h^O outputs the rotation and reflection which is Gram Schmidt orthonormalized, while h^t represent the translation.

3. Experiments

They tested their framework on three tasks: image classification, image segmentation, and point clouds, either on frozen or fine-tuned model.

They confirm that large pre-trained model are not equivariant, and that Prior Regularized Learned Canonicalization (PRLC) can largely address this without the accuracy loss of naive LC. For instance, on CIFAR10, a pretrained Vanilla ResNet50 achieves 96.97% ID accuracy but collapses to 57.77% under $C8$ averaging. On the other hand, PRLC preserves near-equivariance performances (96.19% vs 95.31%). As shown in Tab. 5 (1), PRLC increases (from LC) “identity” canonicalizations from 0.23 to 0.76 on CIFAR10, supporting the alignment explanation (Figs. 3, 4, 5). In segmentation, the canonicalizer capacity is decisive: for COCO Mask RCNN, a small canonicalizer (0.2 M params) yields 35.77 $C4$ -Avg mAP, while a larger one (1.9 M params) reaches 44.50, close to the 45.57 baseline for the non-rotated test set. The main limitations are the residual ID gaps versus Vanilla models, which seem to convey an expressivity bottleneck and reported difficulty scaling the prior to continuous rotations in images. Results tables are provided in the appendix (Figs. 6, 7, 8).

4. Critical Review

4.1. The equivariant problem

Standard neural network architectures, such as CNNs and Vision Transformers, essentially learn a function p . While CNNs are translation equivariance, they do not possess rotation or reflection equivariance. Consequently, the network

needs to learn diverse filters for recognizing vertical edges and horizontal ones despite the figure being geometrically identical but rotated. This require the model to re-learn those features with different orientations, which is computationally expensive if re-training or fine tuning is needed for such models.

Because of this the best way to ensure performance is through higher number of filters, or parameters, which might cause parameter explosion, and scale differently with diverse transformations (cubic with 3D data).

In order to address this problem two methods have been proposed: architectural equivariance and Frame Averaging.

Architectural Equivariance the way in which equivariance is achieved is through restriction to the convolution kernels (filters). For instance, in *Tensor Field Network* (TFNs) (7) we achieve $SE(3)$ equivariance by utilizing spherical harmonics for filter parameterization and Clebsch–Gordan coefficients to couple features, which are represented as a direct sum of $SO(3)$ irreps. While in *General $E(2)$ - Equivariant Steerable CNNs* (5) we generalize 2D convolutions to be equivariant under $E(2)$ rotations and reflections by using steerable kernels, irreps decomposition, and linear constraints.

Since the fundamental operations of such models are different from most others, they often require a training from scratch which is lengthy and costly.

Frame Averaging (FA) also called Group Symmetrization, or the Reynolds Operator, consist of averaging the model’s output over the entire group $\mathcal{G}$. For a non-equivariant model p , an equivariant function f can be constructed by applying the inverse transformation g^{-1} to the output:

$$f(x) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \rho_{out}(g^{-1}) p(\rho_{in}(g) \cdot x)$$

FA (8) forms the theoretical basis of Equi-Tuning (9), which applies this operator to pretrained models in order to minimize the L_2 distance between the original and equivariant features. Reynolds Operator is actually a special solution to a constrained version of the theoretical optimization problem proposed in (9) (more details in App. C) While this enables the use of any backbone, it also cause inference costs to increase proportionally with the cardinality of the group $|\mathcal{G}|$. Moreover, for continuous group, such as $SO(3)$, exact solution are impossible therefore we requires Monte Carlo approximation (10), which might introduce variance and instability.

EquiAdapt addresses the limitations of both approaches by decoupling the equivariance mechanism and avoiding the architectural needs, thus treating the backbone as a black box, while retaining efficiency by executing only a single forward pass. However, while Equi-Tuning guarantees math-

ematical universality and equivariance for discrete groups, EquiAdapt only promises relaxed equivariance.

4.2. Equivariance with Learned Canonicalization (LC)

Underpinning the 'EquiAdapt' approach is the paper (2) that explored the idea of learning equivariance via canonicalization functions.

LC solved intelligently some of the previous problems, but has some core limitations that are addressed by PRLC, for instance in (2) the LC and the backbone are trained jointly, making the misalignment risk impossible to arise.

Here are some clear advantage of using a conditioning prior such as in the presented paper (1):

Scalability because it enables the independent scalability of the backbone, without the need to retrain it. For instance, SAM is comprehensive of 641 million (11), which are not modified while "making it" equivariant.

Versatility the use of an external canonicalization network makes it architecture agnostic, meaning that any specific or desired architecture could be used for the backbone. The authors tested ResNet50, ViT, MaskRCNN, PointNet, and DGCNN. All these have different architectures.

Efficiency by using a canonicalization network we are likely to add only a small overhead in inference time, with respect to complex equivariant architectures and FA. In our paper (1) were added only 0.3% of additional parameters (approx. 1.9M), with increased inference time by a modest 7.3%.

Obviously there are some core limitation to this approach too:

Circular dependency since in order to work EquiAdapt needs the canonicalization function c to be equivariant, then the needs for an equivariant network has simply shifted, but it didn't disappear. We still relies on an equivariant network.

Transformation restrictions the EquiAdapt approach has been applied for well defined equivariant architectures for which established transformation model exist: translation (CNNs), rotation (G-CNN, E(2)-Steerable), and permutations (DeepSets). In particular, experiments were focused on discrete rotations (C_4, C_8) for images and 3D rotations ($SO(3)$) for point cloud. However, it has to be tested on more complex transformations.

Expressivity bottleneck of the canonicalization function highly influence the overall model performance, therefore we might run the risk that its ability to canonicalize might dictate overall task performance. This means that it might requires bigger canonicalization network, falling back into the circular dependency fallacy. For instance, in the COCO dataset experiments using a simple G-CNN (0.2M parameters) resulted in a mAP of 59.28, while the more expressive

G-WideResNet (1.9M parameters) improved mAP to 62.13 (1).

Training time using FA with 100 discretized rotation requires approximately x100 times more at inference, it is also true that there is no need to train an additional network, since the FA's backbone is frozen.

Overall, the EquiAdapt approach seem to have some core advantages, but also some limitations, which might restrict the span of application of this method.

4.3. Continuous groups

Handling continuous symmetries like 2D or 3D rotations can cause significant complications on optimization methodologies. Specifically, the authors reported that enforcing the prior on continuous rotation, using Steerable E(2) networks, was empirically difficult and led to optimization instability (1). Some of the possible problem associated with this might be:

Gradient variance estimating gradients on $SO(3)$ often suffers from high variance, leading to slow convergence.

Topological obstruction canonicalization on $SO(3)$ is impossible for specific objects that have high symmetry, which is why we allow for symmetry breaking, but at the same time this lead to discontinuities in $c(x)$ mapping.

Moreover, the term $\lambda = N(s_q)s_p$ is fixed by the authors, that reduces it to a parameter, but this only works for the specific isotropic case. The computation of the (general) matrix Fisher normalisation constant $n(F)$ involves a hypergeometric function which is computationally expensive and numerically unstable to compute and differentiate (12). This impair a broader generalisation of such approach for continuous rotations.

4.4. Curie's symmetry principle

Curie's principle states that an equivariant function cannot produce an output with less symmetry than its input.

In the paper "Probing Equivariance and Symmetry Breaking in Convolutional Networks" (13) the authors analyze the joint invariance constraint where if X is your data and R is your reference frame (pose), the model $f(X, R)$ must satisfy:

$$f(gX, gR) = f(X, R) \quad \forall g \in G$$

because by providing R the pair (X, R) becomes asymmetric allowing the network to produce specific, non-symmetric outputs, while remaining globally equivariant.

The authors argue that standard invariant models suffer from maximum effective pose entropy, which requires the use of symmetry breaking, through explicit frame reference, to distinguish semantically distinct but geometrically symmet-

ric features. This is supported by the mode failure that is possible to observe in (1), where naive LC result in poor performance due to misalignment. The PRLC function as the practical implementation of this symmetry breaking mechanism. Since by minimizing the KL diverge, EquiAdapt effectively reduce pose entropy to zero.

While (13) internalizes these symmetry breaking in the architecture of *Rapidash* by using geometric input features and global frames, EquiAdapt decouples it through an equivariant canonicalization module. This difference seem to reinforce the idea that symmetry breaking, specifically through prior-regularized canonicalization, help to maintain alignment between the two networks.

4.5. EquiOptAdapt

A core limitation of PRLC is the circular dependency, where in order to avoid larger equivariance network we rely on an additional, yet smaller, equivariant architecture.

Some of the authors addressed this problem with the framework EquiOptAdapt (14). They use the *optimization approach* propose in (2), where we try to minimize the distance (energy) between the input and the canonical sample of the orbit: $s(\rho(g), \mathbf{x})$. Therefore, we can define a canonicalization function as:

$$c(\mathbf{x}) \in \arg \min_{\rho(g) \in \rho(\mathcal{G})} s(\rho(g), \mathbf{x}) \quad (9)$$

where $s : \rho(\mathcal{G}) \times X \rightarrow \mathbb{R}$ can be a neural network and $c(\mathbf{x})$ is a subset of a coset of the stabilizer of $\mathbf{x}$, so it respects eq. 4. Now the equivariance condition on s can be satisfied with a non-equivariant function $u : \mathcal{X} \rightarrow \mathbb{R}$, by defying:

$$s(\rho(g), \mathbf{x}) = u(\rho(g)^{-1} \cdot \mathbf{x}) \quad (10)$$

Therefore, for a discrete group $\mathcal{G}$, we define canonicalization as:

$$g^* \in \arg \min_{g \in \mathcal{G}} u(\rho(g)^{-1} \cdot \mathbf{x}) \quad (11)$$

where $u(\cdot)$ should allow for a unique minimum over the orbit.

Rather than outputting scalars directly, the method learns u through a neural network $s_\theta(\cdot)$ that outputs vectors for each transformed input. Given a reference vector v_R and temperature τ , we can formulate a canonicalization distribution:

$$P_c(g | \mathbf{x}) = \frac{\exp(v_R \cdot s_\theta(\rho(g)^{-1} \mathbf{x}) / \tau)}{\sum_{g' \in \mathcal{G}} \exp(v_R \cdot s_\theta(\rho(g')^{-1} \mathbf{x}) / \tau)} \quad (12)$$

where the canonical element is selected via $\arg \max_g P_c(g | \mathbf{x})$, implemented with a straight-through gradient estimator. Intuitively, v_R converts the high-dimensional embedding vector into a scalar, while τ influence the sharpness and decisiveness of the canonicalization process.

To prevent representation collapse and enforce distinct embeddings across orbit elements, the network is trained to minimize a within-orbit similarity objective, which is effectively a pairwise dot products:

$$\mathcal{L}_{\text{Opt}} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}} \left[\sum_{\substack{g_i, g_j \in \mathcal{G} \\ g_i \neq g_j}} s_\theta(\rho(g_i)^{-1} \cdot \mathbf{x}) \cdot s_\theta(\rho(g_j)^{-1} \cdot \mathbf{x}) \right] \quad (13)$$

where $\mathcal{D}$ is the training dataset. For fine-tuning or zero-shot adaptation an identity prior regularizer is added. With identity e , we formulate the loss as:

$$\mathcal{L}_{\text{prior}} = -\mathbb{E}_{\mathbf{x} \in \mathcal{D}_f} \log P_c(g = e | \mathbf{x}), \quad (14)$$

which encourage canonicalization alignment with the identity.

The results of (14) shows that this approach of training a non-equivariant network to return a score which is then used to determine the canonical transformation to be applied has yield similar or exceeding result when compared to EquiAdapt, both in image classification and segmentation (point cloud was not tested). EquiOptAdapt closed also the gap with baseline models on ID samples (Figs. 10, 11). Additionally, it demonstrated faster converge to canonical form (Fig. 9). Nevertheless, EquiOptAdapt was tested exclusively on C_4 transformation.

In conclusion, this method addresses the circular dependency, increases versatility, improves efficiency, and reduces training time. On the other hand, it still faces an expressivity bottleneck, with a remaining performance gap on ID samples (Tab. 1, (14)), and it is limited to a restricted set of transformations, leaving its generalization to larger groups still an open question.

4.6. Scientific Machine Learning (SciML)

The principles of EquiAdapt could be particularly useful, where symmetries are fundamental physical law, because we could combine the power of fundamental models (e.g. Transformers) to solve complex problem while guaranteeing their equivariant architecture mathematically. This wrapping technique provides the necessary physical inductive biases without the computational burden of redesigning and training specialized equivariant layers throughout the entire internal backbone.

The practical impact of this decoupling is notably evident in $E(3)$ -equivariant problems, like N-body dynamics prediction (Sec. 5.2, (2)). In tasks that aim to predict future positions of interacting particles, standard non-equivariant Graph Neural Networks (GNNs) achieve a MSE of 0.0107. In contrast, a learned canonicalization approach (CN-GNN)

significantly reduces this error to 0.0043 ± 0.0001 (2), outperforming architectures such as Tensor Field Networks (0.0155) and $SE(3)$ Transformers (0.0244).

Furthermore, EquiAdapt offers a scalable alternative to generative modeling frameworks like Action Matching (15). By enforcing geometric symmetries such as $SO(3)$ through an external canonicalization network, we could ensure physical generalization in point cloud tasks, where original implementations may lack such mathematical guarantee. This allows researchers to use the high expressivity of state-of-the-art models while ensuring the final output remains consistent with physical symmetries.

5. Conclusion

This review discussed and analysed EquiAdapt, an approach that shifts from architectural equivariance to a framework that decouples equivariance from prediction tasks via learned canonicalization. Its main contribution is the canonicalization prior, which addresses both the alignment issue and the excessive augmentation effect during training, enabling robust performance without sacrificing ID performance. In comparison with architectural equivariance and Equi-Tuning, or frame averaging, EquiAdapt seems to offer good efficiency and equivariance guarantees by requiring only a single forward pass at inference time and retaining symmetry “control” through the canonicalizer. Nevertheless, it still depends on an equivariant canonicalization network, it has been tested on a limited set of transformations (discrete rotations and point-cloud ones), and it reports optimization difficulties for continuous rotations. Some of these limitations were addressed by EquiOptAdapt, which enables prior-regularized canonicalization even without an equivariant canonicalization network, but is again only explored with a limited scope. Overall, EquiAdapt is a scalable direction, particularly interesting for SciML, due to inherent physical symmetry requirements and the possibility of avoiding expensive retraining or architectural redesign at a time where speed and GPU runtime matter.

References

- [1] Sékou-Oumar Kaba Sai Rajeswar Siamak Ravanbakhsh Arnab Kumar Mondal, Siba Smarak Panigrahi. Equivariant adaptation of large pretrained model. In *Conference on Neural Information Processing Systems*, 2023.
- [2] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.
- [3] Steven Pinker Michael J. Tarr. *Mental rotation and orientation-dependence in shape recognition*, chapter 21(2). Cognitive Psychology, 1989.
- [4] Max Welling Taco S. Cohen. Group equivariant convolutional networks, 2016.
- [5] Gabriele Cesa Maurice Weiler. General $e(2)$ -equivariant steerable cnns. Technical report, University of Amsterdam, 2021.
- [6] Aaron Courville Yoshua Bengio, Nicholas Léonard. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.
- [7] Steven Kearnes Lusann Yang Li Li Kari Kohlhoff Patrick Riley Nathaniel Thomas, Tess Smidt. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [8] Heli Ben-Hamu Ishan Misra Aditya Grover Edward J. Smith Yaron Lipman Omri Puny, Matan Atzmon. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022.
- [9] Karthikeyan Natesan Ramamurthy Vijil Chenthamarakshan Kush R. Varshney Lav R. Varshney Payel Das Sourya Basu, Prasanna Sattigeri. Equi-tuning: Group equivariant fine-tuning of pretrained models. In *Computer Science*, 2023.
- [10] Alex Hernandez Garcia Santiago Miret Fragkiskos D. Malliaros Yoshua Bengio David Rolnick Alexandre Duval, Victor Schmidt. Faenet: Frame averaging equivariant gnn for materials modeling. In *International Conference on Machine Learning*, 2023.
- [11] Nikhila Ravi Hanzi Mao Chloe Rolland Laura Gustafson Tete Xiao Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár Ross Girshick Alexander Kirillov, Eric Mintun. Segment anything. In *Meta AI Research, FAIR*, 2023.
- [12] Alfred Kume Tomonari Sei. Calculating the normalising constant of the bingham distribution on the sphere using the holonomic gradient method, 2013.
- [13] Domas Buracas Christian Shewmake Artem Moskalev Erik Bekkers Sharvaree Vadgama, Mohammad Mohaiminul Islam. Probing equivariance and symmetry breaking in convolutional networks. In *Computer Vision and Pattern Recognition*, 2025.
- [14] Arnab Kumar Mondal Siba Smarak Panigrahi. Improved canonicalization for model agnostic equivariance. In *EquiVision workshop*, 2024.

- [15] Daniel Severo Alireza Makhzani Kirill Neklyudov, Rob Brekelmans. Action matching: Learning stochastic dynamics from samples. In *International Conference on Machine Learning*, 2023.

A. Original paper's results

In this section of the appendix are reported images and tables from the original paper, which are relevant and indicative for the analysis, as well as being cited in the report itself.

Dataset ↓	Model	Training Completed	
		✗	✓
CIFAR10 [31]	Learned Canonicalization (LC) [30]	0.11	0.23
	Prior-Regularized LC (ours)	0.11	0.76

Figure 3. Fraction of images mapped to identity.

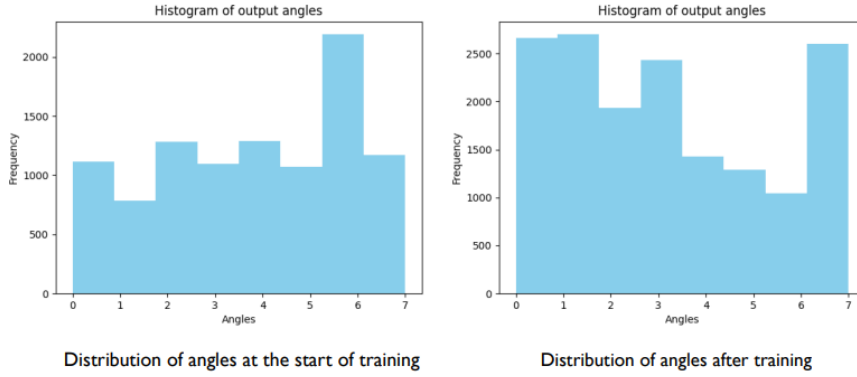


Figure 4. Distribution of angles output from canonicalization function in $C8$ for Learned Canonicalization for CIFAR10 before and after training. We use indices on the x-axis instead of angle values to represent the corresponding multiple of 45 degrees. Frequency denotes the number of images mapped to a particular multiple of 45 degrees.

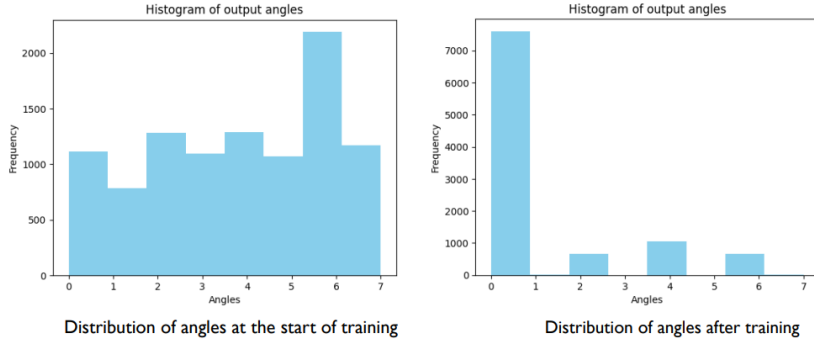


Figure 5. Distribution of angles output from canonicalization function in $C8$ for Learned Canonicalization [30] with prior regularization for CIFAR10 [31] before and after training. We use indices on the x-axis instead of angle values to represent the corresponding multiple of 45 degrees. Frequency denotes the number of images mapped to a particular multiple of 45 degrees.

pretrained Large Prediction Network →		ResNet50		ViT	
Datasets ↓	Model	Acc	$C8$ -Avg Acc	Acc	$C8$ -Avg Acc
CIFAR10 [31]	Vanilla	96.97 ± 0.01	57.77 ± 0.25	98.13 ± 0.04	63.59 ± 0.48
	Rotation Augmentation	94.91 ± 0.07	90.11 ± 0.19	96.26 ± 0.15	93.67 ± 0.39
	Learned Canonicalization (LC) [30]	93.29 ± 0.01	92.96 ± 0.09	95.00 ± 0.01	94.80 ± 0.02
	$C8$ -Aug.	95.76 ± 0.07	94.36 ± 0.09	96.36 ± 0.02	94.17 ± 0.08
	Prior-Regularized LC (ours)	96.19 ± 0.01	95.31 ± 0.17	96.14 ± 0.14	95.08 ± 0.10
CIFAR100 [31]	Vanilla	85.84 ± 0.10	44.86 ± 0.12	87.91 ± 0.28	55.87 ± 0.14
	Rotation Augmentation	80.21 ± 0.09	74.12 ± 0.05	82.59 ± 0.44	78.39 ± 0.89
	Learned Canonicalization (LC) [30]	78.50 ± 0.15	77.52 ± 0.07	80.86 ± 0.17	80.48 ± 0.20
	$C8$ -Aug.	83.00 ± 0.09	79.72 ± 0.10	83.45 ± 0.09	80.08 ± 0.38
	Prior-Regularized LC (ours)	83.44 ± 0.02	82.09 ± 0.09	84.27 ± 0.10	83.61 ± 0.01
STL10 [42]	Vanilla	98.30 ± 0.01	73.87 ± 1.43	98.31 ± 0.09	76.66 ± 0.93
	Rotation Augmentation	98.08 ± 0.06	94.97 ± 0.08	97.85 ± 0.17	94.07 ± 0.11
	Learned Canonicalization (LC) [30]	95.30 ± 0.19	93.92 ± 0.10	95.11 ± 0.01	94.67 ± 0.02
	$C8$ -Aug.	98.31 ± 0.01	96.31 ± 0.13	97.83 ± 0.08	94.45 ± 0.35
	Prior-Regularized LC (ours)	97.01 ± 0.01	96.37 ± 0.12	96.15 ± 0.05	95.73 ± 0.16

Figure 6. Performance comparison of large pretrained models finetuned on different vision datasets. Both classification accuracy and $\mathcal{G}$ -averaged classification accuracies are reported. Acc refers to the accuracy on the original test set, and $C8$ -Avg Acc refers to the accuracy on the augmented test set obtained using the group $\mathcal{G} = C8$. $C8$ -Aug. refers to fine-tuning the pre-trained model with rotation augmentations restricted to $C8$.

Pretrained Large Segmentation Network →		MaskRCNN (46.4 M)		SAM (641 M)	
Datasets ↓	Model	mAP	$C4$ -Avg mAP	mAP	$C4$ -Avg mAP
COCO [22]	Zero-shot (0 M)	45.57	27.67	62.34	58.78
	Prior-Regularized LC [†] (0.2 M)	35.77	35.77	59.28	59.28
	Prior-Regularized LC [‡] (1.9 M)	44.51	44.50	62.13	62.13

Figure 7. Zero-shot performance comparison of large pretrained segmentation models with and without trained canonicalization functions on COCO 2017 dataset. Along with the number of parameters in canonicalization and prediction network, we report mAP and $C4$ -averaged mAP values. [†] indicates G-CNN and [‡] indicates a more expressive G-WRN for canonicalization.

Task →	Classification			Part Segmentation	
Dataset →	ModelNet40			ShapeNet	
Method ↓	z/z	$z/SO(3)$	$SO(3)/SO(3)$	$z/SO(3)$	$SO(3)/SO(3)$
PointNet [3]	85.9	19.6	74.7	38.0	62.3
DGCNN [48]	90.3	33.8	88.6	49.3	78.6
VN-PointNet	77.5	77.5	77.2	72.4	72.8
VN-DGCNN	89.5	89.5	90.2	81.4	81.4
LC-PointNet[30]	79.9 ± 1.3	79.6 ± 1.3	79.7 ± 1.3	73.5 ± 0.8	73.6 ± 1.1
LC-DGCNN[30]	88.7 ± 1.8	88.8 ± 1.9	90.0 ± 1.1	78.4 ± 1.0	78.5 ± 0.9
Ours (with pretrained PointNet and DGCNN for each task)					
	no-aug/ z	no-aug/ $SO(3)$		no-aug/ $SO(3)$	
PRLC-PointNet	84.1 ± 1.1	84.3 ± 1.2		82.6 ± 1.3	
PRLC-DGCNN	90.2 ± 1.4	90.2 ± 1.3		84.3 ± 0.8	

Figure 8. Classification accuracy of different pointcloud models on the ModelNet40 dataset in different train/test scenarios and ShapeNet Part segmentation mean IoUs over 16 categories in different train/test scenarios. x/y here stands for training with x augmentation and testing with y augmentation. z here stands for aligned data augmented by random rotations around the vertical/ z axis and $SO(3)$ indicates data augmented by random 3D rotations

B. Mathematical Proofs

[TO ADD: proof eq. 6 prior loss for continuous rotation, canonicalization is respected for $c(\mathbf{x})$, presentation images and tables. Equi-Tuning (optimization stile format?) **IMPORTANT**

B.1. Equivariance of the canonicalization framework

Referring to 1 in the previous report, we will show equivariance, therefore that $f(\rho(g) \cdot \mathbf{x}) = \rho'(g) \cdot f(\mathbf{x})$, where $f(\mathbf{x})$ is defined in 1.

1. We take any rotation $\mathbf{x}' = \rho(g) \cdot \mathbf{x}$.
2. We substitute $\mathbf{x}'$ into 1:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(c(\rho(g) \cdot \mathbf{x})) \cdot p(\rho(c(\rho(g) \cdot \mathbf{x})^{-1}) \cdot (\rho(g) \cdot \mathbf{x})).$$

3. Given that $c(\mathbf{x})$ is equivariant, i.e. $c(\rho(g) \cdot \mathbf{x}) = \rho(g) \cdot c(\mathbf{x})$, we substitute into the previous expression:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(\rho(g) \cdot c(\mathbf{x})) \cdot p(\rho((\rho(g) \cdot c(\mathbf{x}))^{-1}) \cdot (\rho(g) \cdot \mathbf{x})).$$

4. We use $(a \cdot b)^{-1} = b^{-1} \cdot a^{-1}$, hence

$$(\rho(g) \cdot c(\mathbf{x}))^{-1} = c(\mathbf{x})^{-1} \cdot \rho(g)^{-1}.$$

Substituting:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(\rho(g) \cdot c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x})^{-1} \cdot \rho(g)^{-1}) \cdot (\rho(g) \cdot \mathbf{x})).$$

5. For homomorphism of ρ , $\rho(a \cdot b) = \rho(a) \cdot \rho(b)$, we obtain

$$\rho(c(\mathbf{x})^{-1} \cdot \rho(g)^{-1}) = \rho(c(\mathbf{x})^{-1}) \cdot \rho(\rho(g)^{-1}).$$

Therefore

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(\rho(g) \cdot c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x})^{-1}) \cdot \rho(\rho(g)^{-1}) \cdot (\rho(g) \cdot \mathbf{x})).$$

6. Notice that $\rho(\rho(g)^{-1}) \cdot (\rho(g) \cdot \mathbf{x}) = \mathbf{x}$ (the inverse action cancels the action). Hence the argument of $p(\cdot)$ simplifies to

$$\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x},$$

and we get

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(\rho(g) \cdot c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x}).$$

7. Finally, using that ρ' is also a homomorphism (i.e. $\rho'(a \cdot b) = \rho'(a) \cdot \rho'(b)$), we can factor:

$$\rho'(\rho(g) \cdot c(\mathbf{x})) = \rho'(g) \cdot \rho'(c(\mathbf{x})).$$

Substituting:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(g) \cdot \rho'(c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x}).$$

8. By definition of $f(\mathbf{x})$ in 1,

$$f(\mathbf{x}) = \rho'(c(\mathbf{x})) \cdot p(\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x}),$$

therefore

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(g) \cdot f(\mathbf{x}),$$

which proves equivariance.

B.2. Continuous rotations proposition proof

The cross-entropy is given by

$$\mathbb{E}_{R \sim q} \left[\log p(R \mid \hat{R}_p, s_p) \right] = \int_{SO(n)} q(R \mid \hat{R}_q, s_q) \log p(R \mid \hat{R}_p, s_p) dR, \quad (8)$$

where dR is the invariant Haar measure on $SO(n)$. Here we assume that it is scaled such that

$$\int_{SO(n)} dR = 1.$$

We obtain

$$\mathbb{E}_{R \sim q} \left[\log p(R \mid \hat{R}_p, s_p) \right] = \int_{SO(n)} q(R \mid \hat{R}_q, s_q) \left(s_p \operatorname{Tr} \left[\hat{R}_p^\top R \right] - \log c(s_p) \right) dR, \quad (9)$$

$$= s_p \operatorname{Tr} \left(\hat{R}_p^\top \mathbb{E}_{R \sim q} [R] \right) - \log c(s_p). \quad (10)$$

From Theorem 2.2 and Lemma 2.2 of (?), we have

$$\mathbb{E}_{R \sim q} [R] = \frac{d \log c(s_q)}{ds_q} \hat{R}_q. \quad (11)$$

Therefore, we find

$$\mathbb{E}_{R \sim q} \left[\log p(R \mid \hat{R}_p, s_p) \right] = \frac{d \log c(s_q)}{ds_q} s_p \operatorname{Tr} \left(\hat{R}_p^\top \hat{R}_q \right) - \log c(s_p), \quad (12)$$

which completes the proof.

C. Equi-Tuning

Equi-tuning is a procedure that converts a pretrained (generally non-equivariant) model into a model that is exactly equivariant to a discrete group, while staying as close as possible to the pretrained features on all transformed versions of the inputs.

C.1. Setup

Let $M : X \subset \mathbb{R}^n \rightarrow Y \subset \mathbb{R}^m$ be a pretrained model. Let G be a finite group acting on the input and output spaces via the maps

$$\Gamma_X : G \times X \rightarrow X, \quad \Gamma_Y : G \times Y \rightarrow Y.$$

We write $\Gamma_X(g, x) = gx$ and $\Gamma_Y(g, y) = gy$ when unambiguous. An *equivariant* model M_G satisfies, for all $g \in G$ and $x \in X$,

$$M_G(gx) = g M_G(x). \quad (15)$$

C.2. Equi-tuning objective

Assume the dataset contains all transformations $\{gx : g \in G\}$ for each x , meaning we have a complete orbits under G . The goal is to construct an equivariant M_G that preserves the pretrained representation of M across the orbit. This is formalized by minimizing the discrepancy between pretrained features and the new model features over all group elements:

$$\min_{M_G(x)} \sum_{g \in G} \|M(gx) - M_G(gx)\|_2^2 \quad \text{s.t.} \quad M_G(gx) = g M_G(x) \quad \forall g \in G. \quad (16)$$

To express the loss in a form that depends only on $M_G(x)$, use the equivariance constraint $M_G(gx) = g M_G(x)$:

$$\min_{M_G(x)} \sum_{g \in G} \|M(gx) - g M_G(x)\|_2^2 \quad \text{s.t.} \quad M_G(gx) = g M_G(x) \quad \forall g \in G. \quad (17)$$

If $\Gamma_Y(g, \cdot)$ is an orthogonal (norm-preserving) action on Y (i.e., $\|gy\|_2 = \|y\|_2$), then applying g^{-1} inside the norm yields the equivalent objective

$$\min_{M_G(x)} \sum_{g \in G} \|g^{-1} M(gx) - M_G(x)\|_2^2 \quad \text{s.t.} \quad M_G(gx) = g M_G(x) \quad \forall g \in G. \quad (18)$$

C.3. Closed-form solution via Reynolds operator

A convenient way to solve (18) is to temporarily drop the equivariance constraint, yielding the unconstrained problem

$$\min_{z \in Y} \sum_{g \in G} \|a_g(x) - z\|_2^2, \quad a_g(x) := g^{-1} M(gx). \quad (19)$$

This is a convex least-squares objective in z with unique minimizer given by the mean of the targets $\{a_g(x)\}_{g \in G}$:

$$M_G(x) = \frac{1}{|G|} \sum_{g \in G} g^{-1} M(gx). \quad (20)$$

The operator in (20) is precisely the *Reynolds operator* (group averaging) applied to M .

C.4. Equivariance of the solution

Define M_G by (20). Then M_G is equivariant: for any $h \in G$,

$$M_G(hx) = \frac{1}{|G|} \sum_{g \in G} g^{-1} M(g(hx)) = \frac{1}{|G|} \sum_{g \in G} g^{-1} M((gh)x). \quad (21)$$

Using the change of variables $k = gh$ (a bijection on G since G is a group), we get

$$M_G(hx) = \frac{1}{|G|} \sum_{k \in G} (kh^{-1})^{-1} M(kx) = \frac{1}{|G|} \sum_{k \in G} h k^{-1} M(kx) = h M_G(x), \quad (22)$$

which matches (15). Therefore the closed-form minimizer of the unconstrained problem is already feasible for the constrained problem (18), and hence also solves it.

C.5. Interpretation

Equation (20) is projecting the pretrained model M onto the space of G -equivariant functions by averaging over the group action. This preserves the pretrained knowledge in the sense that minimized the squared deviation from the pretrained features across the full orbit while guaranteeing exact equivariance to G .

C.6. Implementation note

Computing (20) requires evaluating $M(gx)$ for each $g \in G$ and averaging the transformed outputs $g^{-1}(\cdot)$. For finite G , this is straightforward; practical implementations exploit structure in G (and batching) to reduce overhead.

D. EquiOptAdapt

In this section are reported some of the relevant images and results related to the EquiOptAdapt implementation and testing.

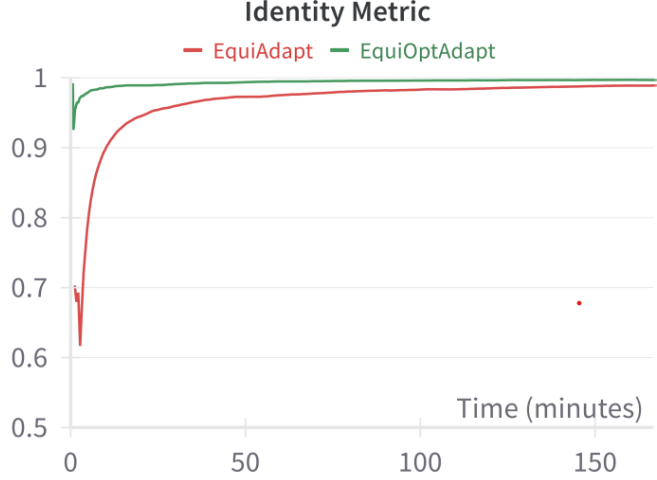


Figure 9. Identity metric vs. Relative wall-time (in minutes). We define the identity metric as the percentage of input images mapped to the identity group element e , which is our prior distribution $\mathbb{P}_{c(x)}$. This figure demonstrates that our EquiOptAdapt is able to learn the prior faster than EquiAdapt.

Pretrained Large Prediction Network →		ResNet50		ViT	
Datasets ↓	Model	Acc	C4-Avg Acc	Acc	C4-Avg Acc
CIFAR10	Vanilla	97.33 ± 0.01	69.72 ± 0.25	98.13 ± 0.04	68.98 ± 0.48
	C4-Augmentation	95.76 ± 0.01	94.77 ± 0.05	96.61 ± 0.04	95.60 ± 0.03
	EquiAdapt	96.19 ± 0.01	96.18 ± 0.02	96.14 ± 0.14	96.12 ± 0.11
	EquiOptAdapt	97.16 ± 0.01	97.16 ± 0.01	96.96 ± 0.02	96.96 ± 0.02
STL10	Vanilla	98.30 ± 0.01	88.61 ± 0.34	98.31 ± 0.09	78.63 ± 0.25
	C4-Augmentation	98.20 ± 0.05	95.84 ± 0.04	97.69 ± 0.07	95.79 ± 0.14
	EquiAdapt	97.01 ± 0.01	96.98 ± 0.02	96.15 ± 0.05	96.15 ± 0.05
	EquiOptAdapt	98.04 ± 0.05	98.04 ± 0.04	97.32 ± 0.01	97.32 ± 0.01

Figure 10. Performance comparison of large pretrained models finetuned on different vision datasets. Both Accuracy (Acc) and C4-Average Accuracy (C4-Avg Acc) are reported. Acc refers to the accuracy on the original test set, and C4-Avg Acc refers to the accuracy on the augmented test set obtained using the group C_4

Network (→)	MaskRCNN		SAM		MaskRCNN	SAM
Setup (↓)	mAP	C4-Avg mAP	mAP	C4-Avg mAP	Inference times (↓)	
Zero-shot	48.19	29.34	62.32	58.77	23m 53s	2h 28m 43s
EquiAdapt	46.80	46.79	62.10	62.10	27m 09s (+13.68%)	2h 34m 36s (+3.96%)
EquiOptAdapt	48.01	48.01	62.30	62.30	25m 35s (+7.12%)	2h 30m 42s (+1.33%)

Figure 11. Zero-shot performance comparison and inference times of large pretrained segmentation models with and without trained canonicalization functions on the validation set of COCO 2017 dataset