Quality Diversity for Visual Pre-Training

Ruchika Chavhan¹, Henry Gouk¹, Da Li², Timothy Hospedales¹,²

¹The University of Edinburgh ²Samsung AI Center, Cambridge
R.Chavhan@sms.ed.ac.uk, {henry.gouk, t.hospedales}@ed.ac.uk, dali.academic@gmail.com

Abstract

Models pre-trained on large datasets such as ImageNet provide the de-facto standard for transfer learning, with both supervised and self-supervised approaches proving effective. However, emerging evidence suggests that any single pre-trained feature will not perform well on diverse downstream tasks. Each pre-training strategy encodes a certain inductive bias, which may suit some downstream tasks but not others. Notably, the augmentations used in both supervised and self-supervised training lead to features with high invariance to spatial and appearance transformations. This renders them sub-optimal for tasks that demand sensitivity to these factors. In this paper we develop a feature that better supports diverse downstream tasks by providing a diverse set of sensitivities and invariances. In particular, we are inspired by Quality-Diversity in evolution, to define a pre-training objective that requires high quality yet diverse features — where diversity is defined in terms of transformation (in)variances. Our framework plugs in to both supervised and self-supervised pre-training, and produces a small ensemble of features. We further show how downstream tasks can easily and efficiently select their preferred (in)variances. Both empirical and theoretical analysis show the efficacy of our representation and transfer learning approach for diverse downstream tasks. Code available at https://github.com/ruchikachavhan/quality-diversity-pretraining.git

1. Introduction

Pre-training neural networks on large-scale datasets such as ImageNet followed by representation transfer is a dominant paradigm in applied deep learning [58]. Supervised pre-training [34] is long established, with self-supervised [14, 30, 15, 11, 69] pre-training rapidly gaining popularity. Such pre-training algorithms often aspire to providing a universal representation which is effective for diverse downstream tasks across computer vision, with evaluation studies showing promising results [34, 24]. Nevertheless, emerging evidence suggests that any single pre-trained feature will not be optimal for diverse downstream tasks [25, 68, 13]. This is because, any given pre-training strategy encodes a certain inductive bias in the learned representation, which will better suit some downstream tasks more than others.

This issue is evident with regards to transformation invariance. By way of example, downstream object recognition benefits from strong affine invariance in a feature since pose is a distractor for object recognition. In con-
trast, downstream pose estimation depends heavily on affine sensitivity in a feature. However, supervised pre-training tends to promote affine-invariance due to the nature of the task and the popular augmentations used. Meanwhile self-supervised pre-training produces feature with even stronger affine-invariance due to optimising for augmentation invariance. So neither may be optimal for downstream pose estimation. More generally, there are a diverse set of potential transformations, each of which may be nuisance factors for some tasks and crucial for others – and any given representation exhibits different different degrees of (in)sensitivity to each each possible transformation. Thus, despite the power of mainstream supervised and self-supervised pre-training, since any single feature induces a particular set of (in)variance strengths, it will be sub-optimal for those downstream tasks with different (in)variance preferences.

Within the self-supervised learning community, a few studies have attempted to address these issues by learning features with multiple (in)variances. This often achieved by an ensemble of different models [25], pre-training multi-headed networks with contrastive learning for different augmentations in each head [12], or with different heads dedicated to (invariance promoting) contrastive learning and (sensitivity promoting) transformation prediction [38]. However, these initial attempts suffer important limitations. Firstly, given that there are many transformations for which invariance or sensitivity may be preferred, it’s unclear how to build a compact enough ensemble. For example, if \( N \) models are used to encode \( N \) distinct strengths of sensitivity to a particular transformation, and there are \( K \) possible transformations, this naively leads to an ensemble of size \( N^K \) to cover the possible invariance preferences of downstream tasks. Secondly, it’s unclear how to optimally select or fuse these different features for downstream tasks with apriori unknown (in)variance preferences.

In this paper, we address the first of these challenges through the lens of Quality Diversity (QD) [12, 51, 27] a strategy in evolutionary computation that has seen great success in robotics [20]. QD strategies aim to more robustly solve problems by searching for a set of solutions rather than a single solution, and – crucially – requiring that the set of solutions are diverse in some meaningful behavioural metric. We introduce the notion of Quality Diversity Optimisation for Vision (QD4V), where, in a computer vision context, we interpret the idea of behaviour space as diversity measurement as degree of (in)variances to different data augmentations. When applied to supervised ImageNet pre-training, this corresponds to optimising for a small set of ImageNet models that are highly performant, yet use as different cues as possible for recognition. This is qualitatively illustrated in Fig. 1. Each downstream task is then presented with a meaningfully diverse set of high-quality features from which a feature meeting the (in)variance preferences of the task is more likely to be found.

Our second contribution addresses how to fuse this small ensemble of features for the downstream task. The limited existing work on learning multiple (in)variances relies on ad-hoc heuristics, such as simply concatenating features prior to linear readout [25, 68]. We argue that a better solution is train a per feature linear readout on the downstream train set, and then learn a fusion weight on the downstream val set, i.e., stacking [66]. We show both empirically and theoretically that this approach to transfer learning outperforms the standard approach.

In summary, we present a framework for learning a compact and high-quality yet meaningfully diverse set of features. We present a simple approach for transferring them to downstream tasks with strong theoretical backing. Finally, we show empirically that our framework improves on average compared to standard pre-trained features when evaluated on a diverse range of downstream tasks.

2. Related Work

Invariance learning in vision: Invariances for in-domain image classification have been learned using data augmentation [9] via MAP [6] and marginal likelihood [32] learning. However, these methods do not focus on transferability to downstream target tasks. On the other hand, the success of representation transfer in contrastive self-supervised learning has been attributed to augmentation-based training engendering invariances [25, 63, 32] which act as strong inductive biases for downstream tasks. The field aspires to provide a single general-purpose feature suited for all downstream tasks [8], however, this goal is not straightforward to achieve. Recent studies have shown that features with invariances to different augmentations are suited for different downstream tasks, with no single feature being optimal for all tasks [25, 68] and performance suffers if inappropriate invariances are provided. In contrastive learning, this leads to the laborious need to produce and combine an ensemble of features [68, 25], to disentangle invariance and transformation prediction [38], costly task-specific self-supervised pre-training [54, 61], or amortising invariances [13]. We introduce a framework for learning diverse invariances during either supervised or self-supervised pre-training, while maintaining the quality of those features in terms of encoding image semantics. We also provide an efficient and theoretically supported approach to fuse the resulting features that improves on standard heuristics [25, 68].

Quality Diversity: Quality Diversity (QD) optimisation [12, 51, 27] originated in evolutionary algorithms as a strategy for finding a meaningfully diverse set of high-performing solutions to a problem, rather than just one global optimum. Access to a suite of meaningfully diverse high-quality solutions often has substantial robustness benefits that are widely exploited in robotics community.
In order to drive diversity, a QD algorithm requires an application-specific measure of behaviour as a feature between which two models’ difference can be compared. We develop a novel behavioural measure for vision in terms of invariances. Mainstream QD optimisation algorithms such as MAP-Elites are not amenable to the differentiable gradient-based solutions required for application in vision. We develop a simple differentiable instantiation of QD that is amenable to easy integration as a loss within a typical deep learning for vision pipeline.

**Ensemble methods:** QD is related to ensemble methods, which have a long history in vision and machine learning [22]. However, ensembles have mostly focused on single-task learning rather than pre-training for transfer learning. A key challenge in ensemble methods is obtaining meaningful inter-model diversity, with the standard approaches of random seeds, boosting, and bagging, providing comparatively weak diversity. Our novel strategy is to take QD’s insight that diversity should be measured in a meaningful space, and identifying (in)variance axes as such a meaningful space for diversity measurement in visual pre-training. A key second issue for ensembles is fusing the ensemble prediction, which has been variously performed by un-weighted prediction averaging [45] or parameter averaging [46], and feature concatenation in transfer learning [25, 68]. We show both empirically and theoretically the efficacy of learning a weighted combination of predictions.

## 3. Methodology

### 3.1 Background

Mainstream QD optimization [27, 51] searches for a diverse set of solutions \( \{\theta_i\} \) to an optimization problem \( \mathcal{L}_q \) by solving an objective like

\[
\min_{\{\theta_i\}} \sum_i \mathcal{L}_q(\theta_i) \quad \text{s.t.} \quad D(\{\pi(\theta_i)\}),
\]

where \( \pi \) is a vector of application relevant observable properties of a given solution \( \theta_i \) that is called *phenotype* in the QD community. \( D \) is a diversity constraint that enforces diversity among the solutions. It is often enforced by tessellation, or gridding, the space of \( \pi \). Thus solution sets \( \{\theta_i\} \) have the maximum possible quality (\( \mathcal{L}_q \)) while also covering the space of measurable differences in behaviour [27, 51].

To make such QD problems more efficient to solve in a vision context, we will relax them to

\[
\min_{\{\theta_i\}} \sum_i \mathcal{L}_q(\theta_i) - \sum_{i \neq j} \mathcal{L}_\text{div}(\pi(\theta_i), \pi(\theta_j)),
\]

where \( \mathcal{L}_\text{div} \) is a loss that measures pairwise distance.

### 3.2 Visual Pre-training

We denote a large dataset for pre-training by \( \mathcal{D}_u \subset \mathcal{X} \times \mathcal{Y} \), where \( \mathcal{X} \) is some space of input images and the space \( \mathcal{Y} \) may be contain labels obtained from manual annotation or pre-text tasks from self-supervision. We introduce a population of \( N \) feature extractors, represented as an ensemble, \( \mathcal{F} = \{f_{\theta_i}\}_{i=1}^N \), which aim to learn a diverse set of invariances while maintaining high quality. Each member of the ensemble is followed by a projection layer \( g_{\theta_i} \), which can be a classifier layer in case of supervised learning or a projection head in case of self-supervised learning [14, 30, 16]. We denote a set of \( K \) augmentations considered for pre-training as \( \mathcal{T} \). Let us also denote a sample \( x \) transformed by augmentation \( \mathcal{T}_j \in \mathcal{T} \) as \( \tilde{x}^j \).

#### 3.2.1 QD for Visual Pre-training: Diversity

To quantify the diversity of solutions during pre-training, we must first define a phenotype — an observable measurement of a feature extractor’s behaviour. We define the behavior of a solution by the degree of invariance it shows to a set of augmentations \( \mathcal{T} \). We borrow the definition of invariance from [25], where invariance of a model to a particular data augmentation is defined as the cosine similarity \( S(f_{\theta_i}(x), f_{\theta_i}(\tilde{x}^j)) \) between augmented and unaugmented features. Thus, we define a phenotype of a solution \( \theta_i \) as a vector \( \pi(x; \theta_i) \in \mathbb{R}^K \) such that \( j^{th} \) element of the phenotype vector indicates the invariance that the model exhibits to augmentation \( \mathcal{T}_j \in \mathcal{T} \), as shown in Eq. 3.

\[
\pi(x; \theta_i) = \left( S(f_{\theta_i}(x), f_{\theta_i}(\tilde{x}^j)) \right)_{j=1}^{\mathcal{T}|}
\]

**Diversity Loss:** Based on the above phenotype definition, we can encourage each model to exhibit a diverse combination of invariances to augmentations in \( \mathcal{T} \). Specifically, we minimise the negative exponential distance between phenotype vector of each member as in Eq 4

\[
\mathcal{L}_\text{diversity} = \sum_{i \neq j} \exp(-|\pi(x; \theta_i) - \pi(x; \theta_j)|)
\]

#### 3.2.2 QD for visual pre-training: Quality

**Quality Loss:** Depending on the pre-training paradigm employed, different loss functions can be used to promote high quality. In the case of supervised learning, the loss function denoted by \( l_q \) can be cross-entropy, while self-supervised methods such as SimCLR/MoCo use contrastive loss [48]. The loss function for the entire ensemble for a sample \( (x, y) \in \mathcal{D} \) is given in Eq. 5

\[
\mathcal{L}_\text{quality} = \frac{1}{N} \sum_{i=1}^N l_q(g_{\theta_i}(f_{\theta_i}(x)), y)
\]

This objective reflects standard supervised or self-supervised learning of the upstream model. Empirically,
we found that purely using this quality objective it is hard for a diverse ensemble to match the quality of a baseline model trained solely for quality. To help the diverse ensemble match the quality of the baseline we introduce an additional loss term, which is the Kullback-Leibler (KL) divergence between the posteriors of the ensemble members. The loss function is shown in Eq. 6. This term ensures that the members of the ensemble mimic the predictions of the baseline model, thus ensuring high quality with diverse invariances.

\[
L_{\text{quality-KL}} = \sum_{i=1}^{N} \frac{1}{m} \sum_{j=1}^{m} \text{KL}(\sigma(\tilde{y}_i; \tau)||\sigma(b_j; \tau))
\]  

(6)

Here, \(\sigma\) denotes the softmax function and \(\tau\) denotes the temperature used to scale the logits within softmax.

**Summary** Finally, to achieve our goal of diverse invariances and high quality in QD4V pre-training, we formulate a total loss function asin Eq. 7.

\[
L_{\text{qdl}} = L_{\text{quality}} + \lambda_{kd} L_{\text{quality-KL}} + \lambda_{d} L_{\text{diversity}}
\]  

(7)

The diverse set of optimal solutions learned from the proposed QD pre-training stage denoted as \(F^*\) is then transferred to a downstream task.

### 3.3. Downstream task Learning

We next address how to construct models for downstream tasks given the diverse ensemble of features \(F^*\) produced by pre-training in the previous section. Existing approaches have concatenated features prior to linear readout [68], or conducted unweighted averaging of ensemble features [45]. We propose to fuse the ensemble via a variant of stacking [66].

**Decoder Design** For each feature extractor, \(f_\theta \in F\), we follow the standard linear readout approach, and build a linear model, \(h^{(i)}\), parameterised by \(U^{(i)} \in \mathbb{R}^{k \times d}\) and \(b^{(i)} \in \mathbb{R}^{k}\).

\[
h^{(i)}(x) = \rho(U^{(i)}f_\theta(x) + b^{(i)}),
\]

where \(\rho\) is a transfer function for linear models (e.g., the softmax function or identity). To produce a final prediction we fuse the ensemble of classifiers \(h^{(i)}(x)\) by learning a weight vector, \(w \in \mathbb{R}^{N}\) that is used to make a linear combination of their predictions,

\[
h(x) = \sum_{i=1}^{N} w_i h^{(i)}(x).
\]

**Decoder Training Strategy** For each downstream task have a collection of data, \(D_{\text{ds}}\), which we split into two non-overlapping subsets, \(D_{\text{tr}}\) and \(D_{\text{val}}\). The hyperparameters of the linear models are tuned using this train–validation split, and the best model parameters found during this tuning phase are denoted by \(U^{(i)}\) and \(b^{(i)}\). Each model is retrained using the full \(D_{\text{ds}}\) dataset using the best hyperparameters to obtain the final values for each \(U^{(i)}\) and \(b^{(i)}\). The procedure used to fit the weights for linearly combining the models is carried out using the \(U^{(i)}\) and \(b^{(i)}\) parameters, and \(w\) is fit on the \(D_{\text{val}}\) set.

\[
w = \arg \min_{w} \mathbb{E}_{D_{\text{val}}} \left[ \sum_{i=1}^{N} v_i \rho(U^{(i)}f_\theta(x) + b^{(i)}) - y \right]^2
\]  

(8)

In the following section, we provide theoretical justification for our proposed downstream architecture and training strategy by way of bounding the generalisation error for downstream tasks learned in this way.

### 3.4. Theoretical Analysis

For ease of exposition, consider the case where \(\rho\) is the identity and the underlying problem is binary classification.
In this case, each \( U^{(i)} \) can be written as a vector, \( \mathbf{u}^{(i)} \). We can define the effective hypothesis class of our algorithm for learning on downstream tasks as

\[
\mathcal{H} = \left\{ \mathbf{x} \rightarrow \sum_{i=1}^{N} w_i(\mathbf{u}^{(i)}, f_{\mathbf{w}}(\mathbf{x})) : \|\mathbf{w} - \mathbb{E}[\mathbf{w}]\|_2 \leq A, \|\mathbf{u}^{(i)} - \mathbb{E}[\mathbf{u}^{(i)}]\|_F \leq B \right\},
\]

where we have absorbed the bias values into \( \mathbf{u} \) by assuming a 1 has been appended to each \( x \). Our goal is to bound the generalisation error of models from this class trained using our stacking procedure in Sec 3.3. The first step is to bound the empirical Rademacher complexity in terms of \( A \) and \( B \) (from the definition of \( \mathcal{H} \)), the norm of the features, and the total amount of data available for the downstream task, \( m = |D_{ds}| \). We then show how the stacking procedure influences the values of \( A \) and \( B \). We use \( \eta \) to indicate the fraction of data used for training, hence \( \eta m = |D_{tr}| \) and \( (1 - \eta)m = |D_{val}| \). Our main result in given in below.

**Theorem 1.** For a model trained using our stacking procedure and the conditions outlined in the statements of Lemmas 1 and 2, we have with probability at least 1 - 2\( \delta \),

\[
\mathbb{E}[\mathbb{1}\{\text{sgn}(h(\mathbf{x})) \neq y\}] \leq \mathbb{E}[|l(h(\mathbf{x}), y)|] + \frac{12X^3\|\mathbf{V}\|_2 \sqrt{\ln(4/\delta)\ln(4N/\delta)}}{\gamma_1\gamma_2 \sqrt{1 - \eta m^{-1/2}}} + \frac{2X^2\sqrt{2\ln(4/\delta)}}{\gamma_1 m} + \frac{2X^2\|\mathbf{V}\|_2 \sqrt{2\ln(4/\delta)}}{\gamma_2\sqrt{T - \eta m}} + 3\sqrt{\frac{\ln(1/\delta)}{2m}},
\]

where \( \mathbb{E}[|l(h(\mathbf{x}), y)|] \) is the mean ramp loss of the model \( h \), computed on the training data.

Notably, this bound achieves the “fast” rate of convergence, \( O(m^{-1}) \), in the complexity terms, with the first term even exhibiting a super-fast rate of \( O(m^{-3/2}) \). This is in contrast to the standard \( O(m^{-1/2}) \) rate in existing bounds. The proof is in the supplemental material.

**Discussion** This theorem depends on both the specific architecture and optimisation strategy outlined in Sec. 3.3, and thus provides justification for these design choices. It says that the generalisation gap between train error and expected (test) error rapidly goes to zero with the size of the downstream train set \( m \). Furthermore, because we have access to an ensemble of features, the train error \( \hat{e}(h) \) can likely be reduced below that of a single feature.

**4. Implementation details**

**Pre-training:** We perform supervised training for ResNet50 and ConvNeXt [42] on ImageNet1K [21] and contrastive training for ResNet50 on both ImageNet1K and ImageNet100 [68]. To learn our feature ensemble, in a parameter efficient way for both architectures, we share the first three layers (layer1, layer2, layer3 according to [65]), after which different members of the ensemble branch out into their own sequence of layers having a separate layer4 and projection head. We initialise the entire ensemble with ImageNet pre-trained models in case of supervised pre-training and MoCo pre-trained models in case of contrastive learning.

**Augmentations:** For supervised pre-training, most methods that rely on strong augmentation policies. Our approach applies the quality loss (Eq. 5) to unaugmented samples only as using augmented samples for quality maximisation would lead to typical strong invariance to all augmentations and prevent diversity. For contrastive learning, we apply very weak augmentations for instance discrimination.

**Learning rates and Optimisers:** For supervised pre-training and contrastive learning experiments, we train the model for 20 epochs with SGD and AdamW optimizers respectively, along with learning rate warm-up for 5 epochs, followed by a cosine decay schedule. For supervised pre-training, we apply label smoothing and perform exponential moving average (EMA) on the model weights.

**Downstream tasks:** Our suite of downstream tasks consists of object recognition, regression, and dense estimation. We provide more details about downstream tasks in the supplementary material.

- **Classification:** For classification, we evaluate on standard benchmarks CIFAR10/100 [36], Caltech101 [26], Oxford Flowers [47], Stanford Cars [35], Describable Textures Dataset (DTD) [17], Aircraft [43].

- **Regression:** We include a set of spatially sensitive tasks including facial landmark detection on 300W [1], CelebA [41], human pose estimation in LSP [33] and MPII [2]. Animal Pose prediction [10], ALOI object orientation prediction [29], and Causal3DIdent 6D pose and appearance attribute prediction [60].

- **Dense estimation:** We also evaluate the QD pre-trained models on dense estimation tasks like Pascal VOC object detection using Faster R-CNN [55] with a Feature Pyramid Network backbone [39] and linear semantic segmentation similar to [4] on CityScapes and ADE20k [7] datasets.

**Downstream evaluation:** For classification and regression tasks, we fit multinomial logistic regression and multi-output linear regression on the extracted features from each
Figure 3. Radar plot comparing invariance w.r.t. for 4 Spatial and 4 Appearance based transformations of baseline models (green) and QD ensemble (dots). The red and blue lines correspond to maximum and minimum invariances within QD ensemble. The blue shaded region corresponds to the range of invariances spanned by the QD ensemble, which encompasses the (in)variance of the baselines.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>Flowers</th>
<th>Caltech 101</th>
<th>DTD</th>
<th>Cars</th>
<th>Aircraft</th>
<th>300w</th>
<th>LS</th>
<th>Pose</th>
<th>Cifar1A</th>
<th>Animal Pose</th>
<th>MPII</th>
<th>ALOI</th>
<th>Insum 3D</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN1K (RN50)</td>
<td>Baseline</td>
<td>90.4</td>
<td>68.2</td>
<td>84.2</td>
<td>94.7</td>
<td>31.9</td>
<td>44.4</td>
<td>36.0</td>
<td>70.2</td>
<td>55.4</td>
<td>49.0</td>
<td>11.2</td>
<td>18.0</td>
<td>24.1</td>
<td>64.1</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>Multitask</td>
<td>90.5</td>
<td>67.3</td>
<td>84.8</td>
<td>94.1</td>
<td>35.0</td>
<td>43.2</td>
<td>35.2</td>
<td>67.3</td>
<td>51.2</td>
<td>49.1</td>
<td>10.9</td>
<td>17.6</td>
<td>22.8</td>
<td>62.3</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>Div. Ens. [23]</td>
<td>91.6</td>
<td>72.9</td>
<td>84.6</td>
<td>89.3</td>
<td>69.8</td>
<td>42.7</td>
<td>35.3</td>
<td>66.6</td>
<td>52.6</td>
<td>49.2</td>
<td>11.5</td>
<td>17.9</td>
<td>22.6</td>
<td>60.2</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>MoCo</td>
<td>91.0</td>
<td>71.8</td>
<td>85.5</td>
<td>85.1</td>
<td>71.8</td>
<td>41.3</td>
<td>34.0</td>
<td>65.3</td>
<td>52.8</td>
<td>47.9</td>
<td>10.5</td>
<td>16.3</td>
<td>23.1</td>
<td>61.2</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>QD4V</td>
<td>90.3</td>
<td>71.4</td>
<td>86.9</td>
<td>89.9</td>
<td>72.2</td>
<td>45.4</td>
<td>36.9</td>
<td>76.6</td>
<td>62.4</td>
<td>61.5</td>
<td>12.5</td>
<td>18.5</td>
<td>28.4</td>
<td>72.8</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>Fusion+ [23]</td>
<td>95.0</td>
<td>71.2</td>
<td>88.2</td>
<td>81.0</td>
<td>86.0</td>
<td>-</td>
<td>-</td>
<td>65.0</td>
<td>68.0</td>
<td>33.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MoCo</td>
<td>91.0</td>
<td>70.6</td>
<td>89.8</td>
<td>81.9</td>
<td>71.9</td>
<td>41.8</td>
<td>37.2</td>
<td>72.2</td>
<td>69.0</td>
<td>49.2</td>
<td>13.9</td>
<td>19.0</td>
<td>46.0</td>
<td>58.1</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>QD4V</td>
<td>88.9</td>
<td>71.2</td>
<td>85.2</td>
<td>88.2</td>
<td>75.8</td>
<td>43.3</td>
<td>43.9</td>
<td>88.9</td>
<td>71.8</td>
<td>95.2</td>
<td>14.2</td>
<td>22.4</td>
<td>47.4</td>
<td>80.6</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 1. Downstream performance of ImageNet (IN1K) pretrained (1) Supervised ResNet50, (2) MoCo ResNet50 and ImageNet-100 (IN100) pretrained (3) Supervised ConvNeXt-Tiny (CN-T), (4) MoCo ResNet50. The first seven columns are classification tasks (accuracy, %); the last seven are regression (R2, %). + numbers from [13] where reported, * a mix of our runs and numbers from [13] where reported, x numbers from [25], rest are our runs.
of the ensemble and then learn a weighted combination of over all these predictions using linear regression. We sweep the regularisation parameters for all members for each downstream dataset based on its validation set. For dense tasks, we train a decoder head for each member of the ensemble and then learn a weighted combination of the predictions by picking among ensemble members.

**Few-shot downstream tasks:** We also evaluate the pre-trained networks on various few-shot learning benchmarks: FC100 [49], Caltech-UCSD Birds (CUB200), and Plant Disease [44]. We also show results for few-shot regression problems on 300w, Leeds Sports Pose, CelebA, and Causal3DIdent datasets, where we repeatedly sampled 5%, and 20% to generate low-shot training sets. For few-shot classification, we perform logistic regression using the frozen ensemble and learn episode-wise fusion weights w the support set itself.

**Competitors:** All methods are evaluated by readouts on fixed features. **Supervised:** Besides conventional supervised pre-training (denoted ‘baseline’), we also train a conventional ensemble via a multi-head model (‘multihead’) that has the same architecture and number of parameters as our architecture but uses only L2 loss. We also compare with another diverse ensemble strategy [23, 50] (denoted ‘div ensemble’): to optimise for diversity in terms of KL between posterior probability of ensemble members. The idea being that differences in secondary probabilities of KL between posterior probability of ensemble members. Finally, we compare our approach to two state-of-the-art ensemble based alternatives AugSelf [68], LOOC [38] and amortised invariances (AI) [13]. Note that unlike other competitors, AI uses backprop to update the features.

<table>
<thead>
<tr>
<th>Methods</th>
<th>50%</th>
<th>75%</th>
<th>CUB200</th>
<th>50%</th>
<th>75%</th>
<th>CUB200</th>
<th>50%</th>
<th>75%</th>
<th>CUB200</th>
<th>50%</th>
<th>75%</th>
<th>CUB200</th>
<th>50%</th>
<th>75%</th>
<th>CUB200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>39.6</td>
<td>81.2</td>
<td>30.4</td>
<td>52.9</td>
<td>48.8</td>
<td>80.3</td>
<td>51.8</td>
<td>3.0</td>
<td>2.8</td>
<td>2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multihead</td>
<td>35.4</td>
<td>80.3</td>
<td>27.3</td>
<td>51.8</td>
<td>49.8</td>
<td>80.2</td>
<td>50.9</td>
<td>3.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QD4V</td>
<td>41.0</td>
<td>81.7</td>
<td>33.8</td>
<td>54.3</td>
<td>50.8</td>
<td>82.5</td>
<td>58.7</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline (FT)</td>
<td>41.0</td>
<td>81.7</td>
<td>33.8</td>
<td>54.3</td>
<td>50.8</td>
<td>82.5</td>
<td>58.7</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QD4V (FT)</td>
<td>65.2</td>
<td>94.3</td>
<td>43.2</td>
<td>67.4</td>
<td>64.2</td>
<td>84.9</td>
<td>62.9</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>40.1</td>
<td>82.0</td>
<td>38.2</td>
<td>54.1</td>
<td>54.0</td>
<td>82.5</td>
<td>58.7</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QD4V</td>
<td>42.9</td>
<td>82.0</td>
<td>39.9</td>
<td>54.6</td>
<td>53.5</td>
<td>82.9</td>
<td>58.1</td>
<td>1.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>35.5</td>
<td>81.7</td>
<td>26.2</td>
<td>56.9</td>
<td>41.2</td>
<td>71.5</td>
<td>42.9</td>
<td>3.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QD4V</td>
<td>37.9</td>
<td>84.9</td>
<td>27.3</td>
<td>40.8</td>
<td>43.3</td>
<td>73.4</td>
<td>48.1</td>
<td>1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Downstream performance of (1) Frozen supervised (2) fine-tuned supervised, (3/4) MoCo trained on ImageNet1K/100 model evaluated on semantic segmentation for Cityscapes and ADE20k datasets, and Pascal VOC object detection.

### 5. Results

**Can we successfully learn diverse (in)variance strengths with QD pre-training?** To answer this question, we use the definition of invariance in Eq. 3 and evaluate the invariances learned by both default models (i.e., supervised and MoCo) and members of our QD ensemble for four appearance augmentations (blur, brightness, grayscale, contrast) and four spatial augmentations (resized crop, horizontal flip, rotation, scale). We compare the invariance strengths learned by the models via the visualization shown in Figure 3. The results show that the standard models provide a fixed set of invariance strengths (green). The members of the QD ensemble (dots) span a diverse range of invariance strengths from low (blue) to high-invariance (red), which can be selected downstream by picking among ensemble members.

**Does QD4V benefit a set of downstream tasks with diverse invariance requirements?** Our suite of downstream tasks is represent a diverse range of objectives with differing invariance-sensitivity requirements. The ability to exploit the appropriate (in)variance for each task is crucial to achieving high performance. As discussed earlier, off-the-shelf models tend towards high-invariance to multiple factors (Figure 3), which is often effective for object...
Freezing the entire backbone and solely train a linear segmentation head and (2) Finetuning the entire backbone. We report the Mean Intersection over Union (IoU) and average accuracy for segmentation. Similarly, we also examine the performance of QD pre-training on Pascal VOC object detection. The results for semantic segmentation and object detection for frozen features and finetuning in Table 3 show that QD pre-training generally provides performance gains.

### Ablation study for decoders

Finally, we compare our proposed downstream task decoder (Section 3.3) with conventional approaches. The results in Table 4 show that learning fusion weights using Eq. 8 consistently performs better, especially for regression tasks than conventional concatenated and prediction averaging. Intuitively, this is because our stacked fusion decoder has the chance for explicit invariance weighting. More technically, this is explained by the fast convergence rate in Theorem 3.4 which shows that our generalisation gap (between train and test error) tends to zero much faster than standard linear models as a function of the amount of training data. (See Supplementary for further discussion).

### How does QD4V compare to large scale pre-trained models?

The seminal CLIP [53] model is highly effective at various downstream tasks. Table 9 (supplementary) compares CLIP/RN50 to ImageNet1K QD4V/RN50 in terms of invariances and performance on downstream pose sensitive tasks. The results show that QD4V substantially outperforms CLIP, despite being trained on orders of magnitude less data - which we attribute to CLIP having learned too strong spatial and appearance invariances. This shows that pre-training scale is not a substitute for providing the correct invariance for the downstream task at hand.

### 6. Conclusion

We addressed the difficulty of pre-training a model to achieve high performance on diverse downstream tasks by introducing the notion of quality-diversity pre-training. By instantiating diversity optimisation in terms of the (in)variance properties of a feature extractor, we obtain a compact set of high-quality features with diverse (in)variance properties. Together with an efficient and effective fusion strategy, we show strong performance on a diverse range of downstream tasks including classification, pose estimation, semantic segmentation, and object detection. Our approach presents a step towards the vision of a universal feature representation capable of supporting multiple vision tasks. In future work we will explore other definitions of behavioural phenotypes in QD4V.

### Acknowledgements

This research was partially supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1 and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Aircraft</th>
<th>Flowers</th>
<th>DTD</th>
<th>Animal Pose</th>
<th>ALOI</th>
<th>Causal3D</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concat</td>
<td>36.8</td>
<td>53.8</td>
<td>85.4</td>
<td>65.6</td>
<td>91.1</td>
<td>21.7</td>
<td>74.8</td>
</tr>
<tr>
<td>Average</td>
<td>33.8</td>
<td>53.8</td>
<td>85.4</td>
<td>65.6</td>
<td>91.1</td>
<td>21.7</td>
<td>74.8</td>
</tr>
<tr>
<td>Ours</td>
<td>36.9</td>
<td>86.9</td>
<td>72.2</td>
<td>12.5</td>
<td>20.4</td>
<td>72.9</td>
<td>1.3</td>
</tr>
</tbody>
</table>

We evaluate MoCo pre-training on ImageNet 1K/100. For classification tasks, we follow [68] in sampling C-way K-shot episodes. For regression tasks we repeatedly sampled 5%, and 20% to generate low-shot training.

### Table 4. Comparison of our proposed downstream task decoder with conventional approaches. Top: Supervised/ImageNet 1k pre-trained. Middle/Bottom: Moco pre-training on ImageNet 1K/100. Our fusion strategy performs better than learning a classifier over concatenated features and averaging features of the ensemble.

Recognition, but may not be effective for fine-grained tasks like CelebA facial attributes, or pose and lighting-prediction tasks like 300W and ALOI/Causal3D.

To evaluate the ability of features to support diverse downstream tasks, we evaluate the competitors on a range of tasks likely to exhibit different invariance preferences in Table 1. Our approach achieves comparable or better performance than baselines on all tasks against both supervised and self-supervised baselines. It has particularly remarkable improvements in regression tasks such as CelebA and Causal3D where we see an improvement of 12.5% and 8.5% over supervised pre-training of ResNet50.

To better understand the success of our method, we analyse the weights assigned to each member of the QD ensemble by different downstream tasks. Specifically, we employ Pearson’s r correlation coefficient between w and the invariances learned by the QD ensemble (in Fig 3) as a metric to measure the strength of the relationship between the two variables. The resulting correlations are shown in Figure 4 and grouped by classification vs regression tasks. The trend shows that the spatially sensitive regression tasks tend to assign lower or negative weights to spatially invariant models, whereas classification tasks prefer models with generally stronger invariances. This analysis illustrates how our framework achieves reliably high performance across the range of tasks evaluated quantitatively in Table 1 – by enabling task-specific invariance preference selection.

### Does QD4V benefit few-shot learning tasks?

To answer this question we focused on MoCo-v2 CNN models trained on ImageNet1K and ImageNet100. For classification tasks, we followed [38, 68] in sampling C-way K-shot episodes from the target problem and training linear readouts and learning w for each episode. For regression tasks we repeatedly sampled 5%, and 20% to generate low-shot training sets. From the results in Table 2, QD pre-training usually performs better than all competitors, with substantial margins in several cases, especially for pose estimation tasks.

### Does QD4V benefit dense estimation tasks?

We evaluate QD pre-training on semantic segmentation on Cityscapes [19] and ADE20k [7] datasets. We evaluate QD4V by (1) Freezing the entire backbone and solely train a linear seg-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean Intersection over Union (IoU)</th>
<th>Average</th>
<th>Concat</th>
<th>Ours</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>37.9</td>
<td>85.9</td>
<td>72.0</td>
<td>12.6</td>
<td>42.9</td>
</tr>
<tr>
<td>Flowers</td>
<td>43.9</td>
<td>85.2</td>
<td>75.9</td>
<td>14.2</td>
<td>57.6</td>
</tr>
</tbody>
</table>

We evaluate MoCo pre-training on ImageNet 1K/100. For classification tasks, we follow [68] in sampling C-way K-shot episodes.
References


[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 8


[26] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR*, 2004. 5, 13


[34] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In CVPR, 2019. 1
[57] Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014. 20
[58] Chunqii Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfeng Liu. A survey on deep transfer learning. In ICANN, 2018. 1