DUET: 2D Structured and Approximately Equivariant Representations

Xavier Suau [1] **Federico Danieli** [1] **T. Anderson Keller** [1 2] **Arno Blaas** [1] **Chen Huang** [1] **Jason Ramapuram** [1] **Dan Busbridge** [1] **Luca Zappella** [1]

Abstract

Multiview Self-Supervised Learning (MSSL) is based on learning invariances with respect to a set of input transformations. However, invariance partially or totally removes transformation-related information from the representations, which might harm performance for specific downstream tasks that require such information. We propose 2D strUctured and approximately EquivarianT representations (coined DUET), which are 2d representations organized in a matrix structure, and equivariant with respect to transformations acting on the input data. DUET representations maintain information about an input transformation, while remaining semantically expressive. Compared to SimCLR (Chen et al., 2020) (unstructured and invariant) and ESSL (Dangovski et al., 2022) (unstructured and equivariant), the structured and equivariant nature of DUET representations enables controlled generation with lower reconstruction error, while controllability is not possible with SimCLR or ESSL. DUET also achieves higher accuracy for several discriminative tasks, and improves transfer learning.

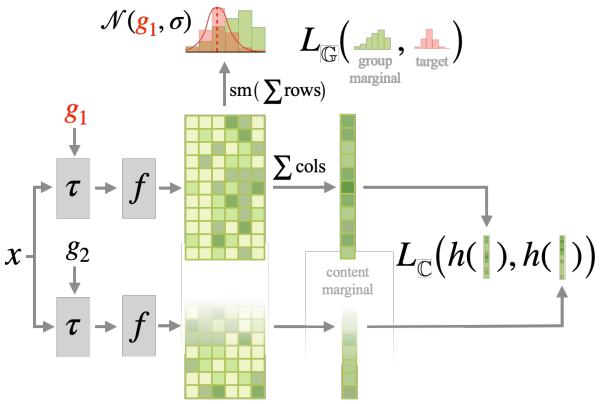
1. Introduction

The field of representation learning has evolved at a rapid pace in recent years, partially due to the popularity of Multiview Self-Supervised Learning (MSSL) (Chen et al., 2020; He et al., 2019; Caron et al., 2020; Grill et al., 2020; Zbontar et al., 2021). The main idea of MSSL is to learn transformation-invariant representations by comparing data views that underwent different transformations. If the transformations alter *only* task-irrelevant information, and if representations of multiple views are similar, then those representations should only contain task-relevant information.

[1]Apple [2]University of Amsterdam. Correspondence to: Xavier Suau <xsuaucuadros@apple.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Figure 1. DUET. The backbone f yields a 2d representation for each transformed image $f(\tau_g(\boldsymbol{x}))$ (*e.g.*, τ_g is a rotation by g degrees). The group marginal is obtained as the softmax (sm) of the sum of the rows, and is compared to the prescribed target (red) with our group loss $L_{\mathbb{G}}$. The content is obtained by summing the columns, and contrasted ($L_{\mathbb{C}}$) with the other view through a projection head h. The final representation for downstream tasks is the 2d one, which has been optimized through its marginals.



However, one can always find a downstream task for which the chosen transformations are relevant. For example, MSSL representations which learn to be color invariant are likely to fail at predicting fruit ripeness where color information is required (Tian et al., 2020), or at the tasks of generation or segmentatation (Kim et al., 2021).

One way to maintain information in the representations is by preserving all possible information from the input, as pursued by InfoMax (Linsker, 1988) frameworks. However, it has been shown empirically and theoretically that for tasks like classification, invariance to nuisance information allows for greater data efficiency and downstream performance (Laptev et al., 2016; Tschannen et al., 2019). In an attempt to simultaneously satisfy the demands of information-rich representations (allowing for generalization to different tasks) and complex invariances (allowing for powerful discriminative representations), modern machine learning research has pursued the concept of structured representations. Colloquially, a representation can be considered structured with respect to a set of transformations if firstly, the transformation between two inputs can be easily recovered by comparing their representations, and secondly, there is a known method

for recovering the representation which is correspondingly invariant to the transformation set. An example of structured representations are convolutional feature maps, which allow for the spatial position (the translation element) to be easily extracted, while similarly allowing for global translation invariance through spatial pooling. Given the success of structured representations, significant work has gone into expanding the range of transformations for which a structured representation can be recovered, for example, rotation, scaling, and other algabraic group symmetries (Cohen & Welling, 2016; Sosnovik et al., 2020; MacDonald et al., 2022; Jiao & Henriques, 2021; Cotogni & Cusano, 2022).

In the context of MSSL, equivariance has also been successfully used to improve distributional robustness (Dangovski et al., 2022; Lee et al., 2021; Keller et al., 2022). However, to date, this equivariance has largely been encouraged at an informational level rather than a structural level, making the careful disassociation of the equivariant and invariant aspects of the representation challenging or impossible. For example, ESSL (Dangovski et al., 2022) and AugSelf (Lee et al., 2021) make representations sensitive to a transformation by regressing the transformation parameter, making their representations theoretically equivariant, but not interpretably structured, as there is no explicit form of the transformation at representation level. Such a lack of structure makes computing invariances or controlled generation from such representations significantly more challenging.

In this work, we present DUET, a method to learn structured *and* equivariant representations with MSSL. Instead of learning 1-dimensional representations as in SimCLR (Chen et al., 2020) or ESSL, DUET representations are reshaped to 2d (see Figure 1). This allows for a richer optimization through their row- and column-wise marginals, which are respectively related to the *group element* (the transformation parameter, *e.g.,* rotation angle) and *content* (all the information that is invariant to the transformation actions). In summary, our main contributions are :

- We introduce DUET, a method to incorporate interpretable structure in MSSL representations for both finite and infinite groups with negligible computational overhead[1]. Our approach also performs well for parameterized transformations that do not satisfy all algebraic group axioms (Serre, 1977), making it widely applicable to most transformations used in MSSL.
- We show empirically that DUET representations become approximately equivariant as a by-product of their predictiveness of a transformation parameter. Importantly, we prescribe an explicit form of transformation at representation level that enables controllable generation, not achievable with ESSL or SimCLR.
- We shed some light on why certain symmetries (*e.g.,* horizontal flips, color transformations) are harder to learn from typical computer vision datasets, due to inherent ambiguity in the data with respect to a transformation. For example, cars appear in both left and right directions, hence making it difficult to define what a *non-flipped* car is.
- We provide extensive experiments on several datasets, comparing with SimCLR and ESSL. We show that DUET representations are suitable for discriminative tasks, transfer learning and controllable generation.

[1]Code available at https://github.com/apple/ml-duet

2. Related Work

Structured and Equivariant Representations. In the unsupervised learning domain, existing works like (Stühmer et al., 2020) have extensively explored structured latent priors for the Variational Autoencoder (VAE) (Kingma & Welling, 2014), while the recent Topographic VAE (Keller & Welling, 2021) aims to induce topographic organization of the observed set of transformations. The idea of structured representations has also been connected to unsupervised learning of disentangled representations (Higgins et al., 2017; Kumar et al., 2018). Another closely related line of work focuses on learning equivariance (Cohen & Welling, 2016; Sosnovik et al., 2020; MacDonald et al., 2022; Jiao & Henriques, 2021; Cotogni & Cusano, 2022) as a more general form of structured representations. For example, (Sosnovik et al., 2020) propose to use a basis of transformed filters to learn equivariant features, which generally leads to improved model robustness and data efficiency. NPTN (Pal & Savvides, 2018) follows on (Sosnovik et al., 2020) and proposes to use a completely learnt basis of filters, learning unsupervised invariances.

Structure in MSSL. Modern MSSL is based on discarding task-irrelevant information via image augmentations. Contrastive and non-contrastive approaches achieve this respectively by comparing augmented views of different data (Chen et al., 2020; He et al., 2019; Caron et al., 2020), or by only comparing views from the same datum (Grill et al., 2020; Zbontar et al., 2021). Several authors have explored the comparison of spatially structured representations (Bachman et al., 2019) (exploiting the InfoMax principle (Linsker, 1988)) or using variants of the NCE (Gutmann & Hyvärinen, 2010) loss (Löwe et al., 2019; Oord et al., 2018; Hjelm et al., 2019). Some works have studied the impact objectives have on the distributions of representations (Wang & Isola, 2020), and how these representations may be identifiable with the latent factors of the data generative process (Zimmermann et al., 2021). Recent works have tackled the preservation of information in MSSL representations. ESSL (Dangovski et al., 2022) supplements SimCLR (Chen et al., 2020) by predicting the parameter of a transformation of choice, and

Table 1. Transformations considered with their associated parameters. Column g shows the corresponding parameters for the group-marginal definition in DUET, and *Target* shows the recommended target distribution. Note that flips are mapped to $\{\frac{1}{4}, \frac{3}{4}\}$, turning them into cyclic groups.

Transform.	Finite	Parameter	g	Target
Rot. (4-fold)	$\surd$	$\{0, 90, 180, 270\}$	$\{\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}\}$	vM
Rot. (360)		$[-180, 180]$	$[0, 1]$	vM
H. Flip	$\surd$	$\{0, 1\}$	$\{\frac{1}{4}, \frac{3}{4}\}$	vM
V. Flip	$\surd$	$\{0, 1\}$	$\{\frac{1}{4}, \frac{3}{4}\}$	vM
Grayscale	$\surd$	$\{0, 1\}$	$\{0, 1\}$	$\mathcal{N}$
Brightness		$[0.6, 1.4]$	$[0, 1]$	$\mathcal{N}$
Contrast		$[0.6, 1.4]$	$[0, 1]$	$\mathcal{N}$
Saturation		$[0.6, 1.4]$	$[0, 1]$	$\mathcal{N}$
Hue		$[-0.1, 0.1]$	$[0, 1]$	$\mathcal{N}$
RRC		$[0.2W, W]$	$[0, 1]$	$\mathcal{N}$

obtaining theoretically equivariant representations. Similarly, although not focused on equivariance, AugSelf (Lee et al., 2021) predicts the difference in transformation parameters between two views. PCL (Li et al., 2020) adds a reconstruction loss to preserve information about the input. Concurrent work (Huang et al., 2023) disentangles the feature space with masks learned via augmentations.

3. Preliminary Considerations

3.1. Groups and Equivariance

Let $f : \mathbb{X} \mapsto \mathbb{Z}$ be a mapping from data to representations. Such mapping is equivariant to the algebraic group $\mathcal{G} = (\mathbb{G}, \cdot)$ if there exists an input transformation $\tau : \mathbb{G} \times \mathbb{X} \mapsto \mathbb{X}$ (noted $\tau_g(\boldsymbol{x})$) and a representation transformation $T : \mathbb{G} \times \mathbb{Z} \mapsto \mathbb{Z}$ (noted $T_g(\boldsymbol{z})$) so that

$$T_g(f(\boldsymbol{x})) = f(\tau_g(\boldsymbol{x})), \qquad \forall g \in \mathbb{G}, \qquad \forall \boldsymbol{x} \in \mathbb{X}. \tag{1}$$

If τ_g and T_g form algebraic groups in the input and representation spaces respectively, then the mapping f preserves the structure of the input group in the representation space (homomorphism). Recall that for $(\mathbb{G}, \cdot)$ to form a group, the properties of *closure*, *associativity*, and existence of *neutral* and *inverse* elements must be satisfied (Serre, 1977). Here we consider both finite and infinite groups.

3.2. On MSSL Input Transformations

In MSSL, τ_g is defined by a parameterized transformation applied to the input. For example, rotation is parameterized by a real angle ($g \in \mathbb{R}$, infinite group). If $g \in [0, 2\pi]$ then it forms a cyclic group. One can also use discrete rotations which form a finite group where $g \in \{0°, 90°, 180°, 270°\}$. However, not all input transformations form a group. For example, a change in image contrast moves some pixel values out of range, thus clipping is applied which invalidates the *associativity* property (*e.g.,* $\tau_{2.0}(\tau_{0.5}(\boldsymbol{x})) \neq \tau_{0.5}(\tau_{2.0}(\boldsymbol{x}))$). We also include RandomResizedCrop (RRC) in our study using the relative cropped width W as a proxy for scale (assuming loss of information about location and aspect ratio). All transformation parameters are mapped in $[0, 1]$ by min-max normalization as shown in Table 1. Although some transformations do not form a group (*e.g.,* RRC, color transformations), the concept of equivariance is often relaxed to embrace transformations which do not form groups. Note that this assumption does not invalidate our methodology for exact groups, and helps understand how our method is suitable for non-exact groups.

4. DUET Representations

In this section we describe how we can use DUET to learn representations that are structured with respect to an algebraic group $\mathcal{G} = (\mathbb{G}, \cdot)$. The overall DUET architecture is shown in Figure 1. A training input image $\boldsymbol{x}$ is transformed twice by sampling 2 group actions from the same group $g_1, g_2 \in \mathbb{G}$ (*e.g.,* two angles of rotation). We obtain the transformed images $\boldsymbol{x}_k = \tau_{g_k}(\boldsymbol{x})$ with $k = 1, 2$. Let f be a deep neural network backbone such that $\boldsymbol{z}_k = f(\boldsymbol{x}_k) \in \mathbb{R}^{C \times G}$, where C and G are the number of rows and columns in the representation, as shown in Figure 1. This 2-dimensional representation $\boldsymbol{z}_k$ models the joint (discretized and unnormalized) distribution $P(\mathrm{c}, \mathrm{g}|\boldsymbol{x}_k)$ where $\mathrm{c} \in \mathbb{R}^C$ and $\mathrm{g} \in \mathbb{G}$ are two random variables defined in the content and group element domains. Our joint interpretation allows to marginalize $P(\mathrm{c}|\boldsymbol{x}_k)$ by summing the columns of $\boldsymbol{z}_k$, and $P(\mathrm{g}|\boldsymbol{x}_k)$ by summing the rows. Rather than imposing a certain dependence (or independence) structure between c and g, (conditioned on $\boldsymbol{x}_k$), we only impose our objectives on the marginals $P(\mathrm{c}|\boldsymbol{x}_k)$ and $P(\mathrm{g}|\boldsymbol{x}_k)$ and let the model learn such dependencies from the data. Note that a final Batch Normalization (BN) (Ioffe & Szegedy, 2015) layer in f will make the mean of $\boldsymbol{z}_k$ to be approximately β (bias term in BN). This is important for equivariance as shown in Section 4.5. Although we focus on a single group, DUET's formulation is suited to handle multiple groups as discussed in Appendix C, which we leave as future work.

4.1. The Group Marginal Distribution

As we marginalize $\boldsymbol{z}_k$ over the content dimension (C) we get $\{\mu_j\}_{j=1}^{G}$, the sum of each column in $\boldsymbol{z}_k$. We obtain our discretized group marginal $P(\mathrm{g}|\boldsymbol{x}_k)$ by softmaxing μ_j. Since the parameters g_k sampled during training are known, we can design a target distribution for $P(\mathrm{g}|\boldsymbol{x}_k)$ (red distribution in Figure 1). We use a von-Mises (vM) target $q(\mathrm{g}|\boldsymbol{x}_k) = \mathrm{vM}(g_k, \kappa)$ for cyclic groups, and a Gaussian ($\mathcal{N}$) target $q(\mathrm{g}|\boldsymbol{x}_k) = \mathcal{N}(g_k, \sigma)$ for all other groups. Both

targets are chosen because of the simplicity by which we can encapsulate parameter information in their structure (their mean), and the controllability of the uncertainty about g via σ (or κ). For readability, we refer to the uncertainty as σ for both vM and $\mathcal{N}$ targets, where $\sigma \approx \frac{1}{\sqrt{2\pi\kappa}}$.

To be comparable to $P(\mathrm{g}|\boldsymbol{x}_k)$, we also discretize our target in $[0, 1]$ obtaining $Q(\mathrm{g}|\boldsymbol{x}_k)$. Let Ω_j be the intervals of a G-sized partition, and g_j their centers. Then, the discretized target is obtained by integrating the continous target according to the partition: $Q_j(\mathrm{g}|\boldsymbol{x}_k) := Q(\mathrm{g} = g_j \,|\, \boldsymbol{x}_k) = \frac{\int_{\Omega_j} q(\mathrm{g}|\boldsymbol{x}_k)dg}{\int_0^1 q(\mathrm{g}|\boldsymbol{x}_k)dg}$. For the Gaussian target, we assume a slight boundary effect as we do not integrate the tails beyond Ω_j.

We encourage the observed $P(\mathrm{g}|\boldsymbol{x}_k)$ to match the target by minimizing the Jensen-Shannon Divergence (D_{JS}) between the discretized distributions. The group loss for the i-th image $\boldsymbol{x}^{(i)}$ in a batch is

$$L_{\mathbb{G}}^{(i)} = \frac{1}{2}\sum_{k=1}^{2} D_{\mathrm{JS}}\big(P(\mathrm{g}|\boldsymbol{x}_k^{(i)}) \parallel Q(\mathrm{g}|\boldsymbol{x}_k^{(i)})\big). \tag{2}$$

The choice of σ is key to encourage structure Both very small and very large values of σ will lead to a loss of structure in the columns of $\boldsymbol{z}$. For small σ, the target takes a form close to a δ distribution. This results in an invariant discretized target (as δ moves inside interval Ω_j) or abrupt target changes (as δ moves from Ω_j to Ω_{j+1}), which prevents learning proper structure. Conversely, for large σ, the target will be close to a uniform distribution, thus removing all information about the group element (all columns contribute equally). In Appendix H.1 we find empirically that $\sigma \approx 0.2$ is optimal in our setting. Note that this value corresponds to a normal distribution $\mathcal{N}(\cdot, 0.2)$ that covers the $[0, 1]$ domain within approximately its 3σ span (when centered at 0.5), being a good trade-off in terms of structure. Interestingly, since our group elements are bounded in $[0, 1]$, the value of σ can be kept constant for all transformation groups and data sets.

4.2. The Content Marginal Distribution

As we marginalize $\boldsymbol{z}_k$ by summing over the group dimension (G), we obtain $P(\mathrm{c}|\boldsymbol{x}_k)$, the probability of observing the content c given $\boldsymbol{x}_k$. Such distribution is invariant to the group actions, and contains all relevant information of $\boldsymbol{x}_k$ not related to the group $\mathbb{G}$. For example, the content of an image of a horse is still a horse regardless of is rotation. We maximize the agreement between the content of two views of an image $(\boldsymbol{x}_1, \boldsymbol{x}_2)$. Our content representation is defined directly by the values of $P(\mathrm{c}|\boldsymbol{x}_k)$, noted as $\boldsymbol{c}_k \in \mathbb{R}^C$. Following the recent trends in MSSL, both content representations are projected with a network h. Then we use the NTXent loss (Chen et al., 2020) in form of $L_{\mathbb{C}} = \mathrm{NTXent}(h(\boldsymbol{c}_1), h(\boldsymbol{c}_2))$ for a SimCLR-based architecture.

4.3. The DUET Loss

Our final loss for a full batch of N images is

$$L_{\mathrm{DUET}} = \frac{1}{N}\sum_{i=1}^{N} L_{\mathbb{C}}^{(i)} + \lambda L_{\mathbb{G}}^{(i)}. \tag{3}$$

$L_{\mathbb{C}}$ encourages similarity between the content representations of 2 views, explicitly *made* invariant to the group action, as opposed to SimCLR which contrasts representations *to achieve* invariance to the group action. The parameter λ controls how strongly the group structure is imposed.

4.4. Recovering the Transformation Parameter

An interesting property of DUET representations is the ability to recover the transformation parameter of a test image without relying on extra regression heads. This property is useful to transform representations equivariantly (see Section 4.5). It also enables interpretability, since one can analyze the default transformation parameters associatied to an image or a dataset. Assuming optimal training of $L_{\mathbb{G}}$, the group marginal will resemble the imposed target. Therefore, the transformation parameter $\tilde{g}$ of an arbitrary image $\boldsymbol{x}_k$ for a Gaussian target is directly recoverable as $\tilde{g} = \mathbb{E}[\mathrm{g}|\boldsymbol{x}_k] \approx \sum_{j=1}^{G} P_j(\mathrm{g}|\boldsymbol{x}_k) g_j$. In practice, for improved robustness, we fit a Gaussian (or vM) function to the values $P_j(\mathrm{g}|\boldsymbol{x}_k)$, and we estimate $\tilde{g}$ as is argmax.

4.5. Equivariance in DUET

Similarly to the approach of ESSL, DUET encourages equivariance by making the neural network sensitive to the transformation parameter g. However, in our method this sensitivity is defined explicitly via Equation (2), such that applying a transformation $\tau_g(\boldsymbol{x})$ in input space results in shifting by g the mean of the group marginal distribution corresponding to their representation $\boldsymbol{z} = f(\boldsymbol{x})$. In practice, we prescribe *a-priori* a form for the feature-space transformation T_g corresponding to the input-space transformation τ_g in Equation (1), with the advantage of gaining additional controllability over such transformations (see also Section 5.2).

Specifically, we design T_g according to the following considerations. Assuming optimal training of $L_{\mathbb{G}}$, we have that the recovered group marginal distribution $P(\mathrm{g}|\boldsymbol{x}_k)$ for a given input $\boldsymbol{x}$ transformed by τ_{g_k} resembles the target $Q(\mathrm{g}|\boldsymbol{x}_k)$. For Equation (1) to hold, we need to design T_g such that applying the column sum and softmax operations used to derive $P(\mathrm{g}|\boldsymbol{x}_k)$ (see Section 4.1) to $T_{g_k}(\boldsymbol{z})$ also resembles $Q(\mathrm{g}|\boldsymbol{x}_k)$. We can ensure this by changing the column sums of $\boldsymbol{z}$ (denoted as $\{\mu_j\}_{j=1}^{G}$) with values $\{\hat{\mu}_j\}_{j=1}^{G}$ that after applying the softmax yield $Q(\mathrm{g}|\boldsymbol{x}_k)$, i.e. $\hat{\mu}_j = \mathrm{softmax}^{-1}(\hat{Q}_j)$ with $\hat{Q}_j = Q_j(\mathrm{g}|\boldsymbol{x}_k)$ for ease of no-

tation. There are infinitely many solutions for $\hat{\mu}_j$, so we choose the $\hat{\mu}_j$ that satisfies $\sum_j \hat{\mu}_j = \beta_j$. This choice comes from the fact that the final BN layer in f will make the mean of $\boldsymbol{z}$ close to the BN bias terms β_j.

Therefore,

$$\hat{\mu}_j = \ln \hat{Q}_j + \ln \sum_j e^{\hat{\mu}_j} \quad \text{with} \quad \sum_j \hat{\mu}_j = \beta_j. \tag{4}$$

The solution to this equation is given by

$$\hat{\mu}_j = \ln \hat{Q}_j + \beta_j - \frac{1}{G} \sum_j \ln \hat{Q}_j. \tag{5}$$

Finally, we define T_g so that it swaps the mean μ_j with $\hat{\mu}_j$

$$T_g(\boldsymbol{z}) = \boldsymbol{z} - \boldsymbol{M} + \widehat{\boldsymbol{M}}_g, \tag{6}$$

where all elements of each column j of $\boldsymbol{M}$ (or $\widehat{\boldsymbol{M}}_g$) take the value μ_j (or $\hat{\mu}_j$). As such, applying the column sum and softmax operations to $T_{g_k}(\boldsymbol{z})$ yields the same values as applying them to $\boldsymbol{z}_k$ (at optimality), which is a necessary condition for Equation (1) to hold. Furthermore, defined in this way, T_g satisfies the group axioms (Appendix B, again assuming $L_{\mathbb{G}}$ is minimized).

In practice, as it also happens in other works such as ESSL or TVAE, we cannot expect Equation (1) to hold always (i.e. for all $\boldsymbol{x}$ and g), as that would require perfect generalization of the learned equivariance. However, for our method, we can bound the equivariance generalization error w.r.t. unseen g (Appendix A), and furthermore demonstrate that it is small in practice in Section 5.1.

On predictiveness and equivariance. It is key to differentiate between predictiveness and equivariance. While predictiveness implies equivariance, the opposite is not always true (*e.g.,* invariance is a specific case of equivariance that does not imply predictiveness). Therefore, we emphasize that the approximate equivariance in DUET is a by-product of the predictiveness of g at group marginal level.

5. Experimental Results

5.1. Empirical Proof of Equivariance

We start with an empirical validation of equivariance by measuring how Equation (1) holds for real data. To do this, we use the transformation T_g from Equation (6) and compute representations $f(\tau_{g_1}(\boldsymbol{x}))$ and $T_{g_2}(f(\boldsymbol{x}))$ $\forall g_1, g_2 \in \mathbb{G}$. An equivariant map should result in a minimal L2 distance $\ell_{g_1,g_2} = \|f(\tau_{g_1}(\boldsymbol{x})) - T_{g_2}(f(\boldsymbol{x}))\|_2^2$ when $g_1 = g_2$. To verify this, we plot the pairwise ℓ_{g_1,g_2} for all elements g_1, g_2 and different transformations. More precisely, we sweep 100 values of g_1, g_2 in $[0, 1]$ for 1000 randomly selected CIFAR-10 (Krizhevsky, 2009) test images and we show the average pairwise L2 distance in Figure 2.

For infinite groups (*i.e.* color transformations and rotation (360)), there is a strong similarity along the diagonal, validating Equation (1). For finite groups (rotation 4-fold, flips and grayscale) we also see a strong similarity at the observed group elements. For example, rotation 4-fold shows 4 minima at the observed (normalized) angles. These plots also help to understand how the model generalizes to unseen group elements. Interestingly, equivariance for horizontal flip is only mildly learnt due to its ambiguity in the dataset (see Section 6 for extended discussion). Indeed, flipped images appear naturally in CIFAR-10 (*e.g.,* cars looking to the right or left), and thus there is more ambiguity about the meaning of image flipping. Vertical flips are nicely learnt, since they do not naturally appear in data. Another interesting observation is that grayscale yields a constant representation as we reduce the saturation (horizontal axis) and then shows a sudden jump close to 1 (grayscale image). The model has learnt that, as soon as the image presents *some* hint of color, it is *not* grayscale, unless it is *purely* grayscale. Note also that grayscale does not form an algebraic group, yet DUET is still able to learn its structure.

5.2. DUET Representations for Group Conditional Generation

In Figure 3 we showcase the benefit of equivariance in DUET representations to conditioning generation on specific group elements. To this end, we train a decoder on frozen pre-trained DUET representations. In this work we do not aim to obtain state-of-the-art generation quality, but rather use a decoder for visual validation of our hypotheses. Note that group conditional generation is not feasible with MSSL methods like SimCLR or ESSL since there is no explicit transformation at representation level.

Here we exploit the equivariant property of DUET representations for controlled generation. We first obtain the representation of a test image $\boldsymbol{z} = f(\boldsymbol{x})$ (leftmost images in Figure 3), then we create multiple transformed representations $\{T_g(\boldsymbol{z})\}$ using Equation (6), by sweeping g between 0 and 1, and finally we decode all $\{T_g(\boldsymbol{z})\}$. In Figure 3 we show the decoded images for different datasets and groups. Notice how we can recover the input transformation by only transforming the representations, which provides yet an additional visual proof of equivariance in DUET. In Appendix F we show that the reconstruction error of DUET is up to 66% smaller than with SimCLR (rotation (4-fold)) and up to 70% smaller than with ESSL (grayscale).

Figure 2. Empirical validation of equivariance in DUET. We measure the L2 distance between the representations of a transformed image $f(\tau_g(\boldsymbol{x}))$ and the transformed representations of that image $T_g(f(\boldsymbol{x}))$, varying $g \in [0, 1]$ along both axes. The plots show the average L2 distance for 1000 CIFAR-10 test images. Note the strong similarity for the same group element (diagonal), and the cyclic nature of rotations or flips when using a vM target (**top row**), as opposed to the Gaussian target (**bottom row**). More results in Appendix E.

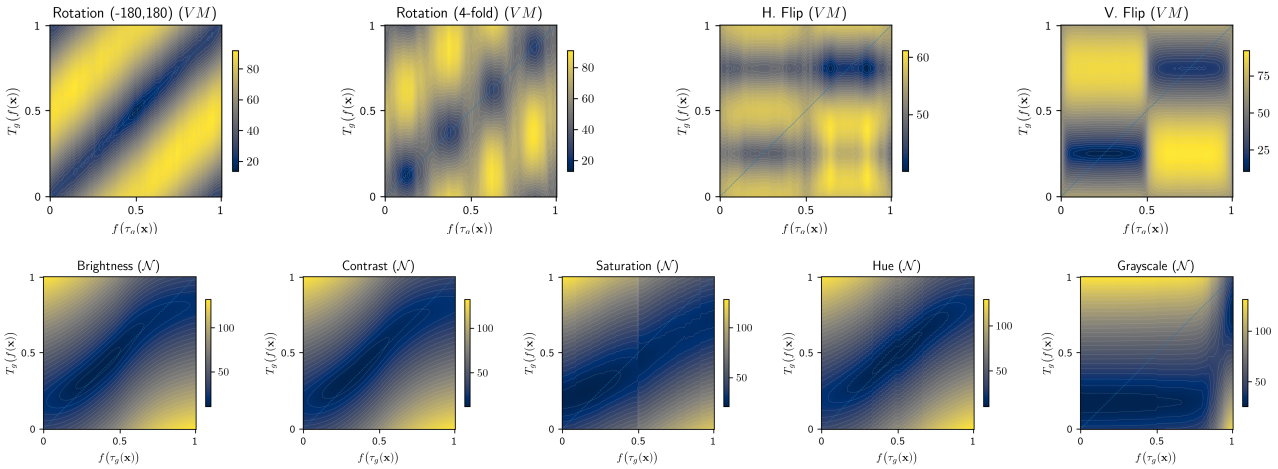
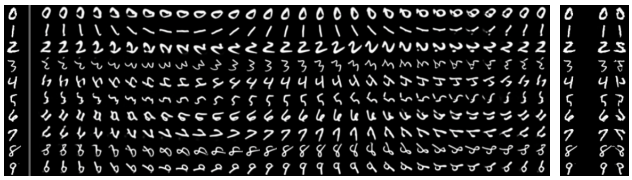


Figure 3. Equivariance in DUET. We encode a test image (leftmost images), transform its representation using T_g (Equation (6)) for several g, and then decode the transformed representations. See how transforming the representations exposes the input transformation learnt by the model, empirically proving equivariance.



(a) MNIST with *Rot. (360)* (left) and *horizontal flip* (right).

(b) CIFAR-10 for *Rot. (360)* and *color* transformations.

5.3. DUET Representations for Classification

5.3.1. RRC+1 Experiments

In this section we analyze how DUET representations perform for discriminative tasks. Following the procedure in the ESSL work, where a single transformation is applied on top of RandomResizedCrop (RRC), we carry out the set of RRC+1 experiments. We compare our method with SimCLR and ESSL[2]. We also compare with a variant of our method (coined DUET$_{\lambda=0}$) optimized without the group loss, that is with $\lambda = 0$ in Equation (3). Notice that in DUET$_{\lambda=0}$ we still reshape the features to 2d and sum over the columns to obtain the content representation (that is contrasted), which is a fundamental difference with SimCLR. DUET$_{\lambda=0}$ learns unsupervised invariances very similarly to what NPTN (Pal & Savvides, 2018) does, but does not guarantee equivariance. For RRC+1, DUET uses $\lambda = 10$, except for *rotations* and *vertical flip* for which we use $\lambda = 1000$ according to the empirical study in Appendix H.1. The remaining parameters are set to $\sigma = 0.2$ and $G = 8$. The full training procedure is provided in Appendix H.

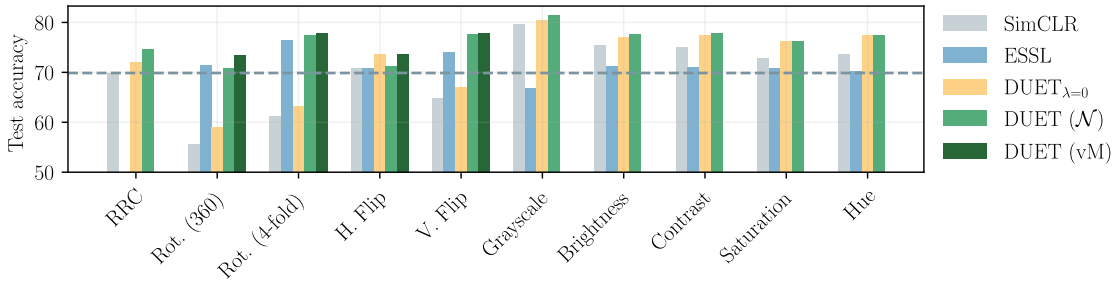
In Figure 4 we show the accuracy of a linear tracking head for the RRC+1 experiment on CIFAR-10. The horizontal dashed line shows the baseline performance of SimCLR with only RRC. For all considered transformations, we show results from training with a $\mathcal{N}$ target group-marginal. For cyclic transformations (rotations and flips), we further specialize the target and consider a (periodic) vM distribution instead, reporting results for this case also. It is important to point out that, for DUET, the tracking head receives our 2d representation flattened, and as such is of the *same dimensionality* as in compared methods.

Our method outperforms SimCLR for all transformations, and even improves over SimCLR with RRC only by learning structure with respect to scale. Note that, by construction, ESSL cannot improve over the RRC-only baseline. A prominent result is the performance of DUET with color transformations. For the discrete transformation *grayscale*, ESSL degrades performance by 12.8% with respect to SimCLR with grayscale, while DUET improves it by 1.75%. For continuous color transformations, DUET improves over SimCLR between 3-5%, while ESSL degrades the performance by up to 4.3% (brightness). This shows that the implicit equivariance in ESSL is not sufficient in this case.

We also observe in Figure 4 that ESSL does not improve

[2]ESSL representations are implicitly equivariant but do not guarantee interpretable structure with respect to the transformation.

Figure 4. Test top-1 performance of a linear tracking head on CIFAR-10. It can be observed that DUET improves over SimCLR and ESSL for all transformations. Notably for continuous color transformations, ESSL significantly degrades performance unlike DUET.



over SimCLR for horizontal flips, while DUET (vM) improves by 2.8%. In general, for ambiguous transformations like horizontal flip (see Section 6), we find that learning unsupervised structure with DUET$_{\lambda=0}$ is beneficial. We also see that structure (and equivariance) is strongly helpful for vertical flips. For the more complex cyclic transformations, DUET$_{\lambda=0}$ underperforms by a large margin, since such complex structure is harder to learn in a completely unsupervised way. This result shows that accounting for the topological structure of the transformation (as studied by Falorsi et al. (2018)) is of great importance, and opens the door to further research in this direction. Surprisingly, DUET$_{\lambda=0}$ outperforms SimCLR. We speculate that the unsupervised structure learnt by DUET$_{\lambda=0}$ might induce a more discriminative organization of the embedding space.

Table 2 benchmarks the more complex tasks CIFAR-100 (Krizhevsky, 2009) and TinyImageNet (Li et al., 2017). We report the average across the cyclic groups and the color-related groups for better readability. DUET achieves the highest accuracy compared to all algorithms tested, including DUET$_{\lambda=0}$, and across all groups but horizontal flip. DUET also improves under color transformations with respect to SimCLR with the same transformations, while ESSL shows a degradation. Indeed, the datasets used in Table 2 present higher data scarcity per class than CIFAR-10. In such setting, the structure learnt by DUET shines over unstructured methods like ESSL.

5.3.2. Full Augmentation Stack Experiments

In this section we use the full augmentation stack as in SimCLR (see details in Appendix G). We learn structure for one group at a time, while applying the full stack on input images. Note that in the full stack setting, we use a fixed $\lambda = 10$ for DUET[3]. We observed in this case that extremely large λ can harm performance since multiple transformations add ambiguity to the group being learnt.

[3]We did not perform an extensive hyper-parameter tuning, the focus of this work being an exploration of structured representations in MSSL.

Table 3 reports the test top-1 accuracy on CIFAR-10, CIFAR-100 and TinyImagenet. One interesting observation is that DUET becomes better than the compared methods as the dataset complexity increases, achieving the best average accuracy for all sets of transformations on TinyImageNet. For smaller and simpler datasets like CIFAR-10, DUET outperforms ESSL for color transformations, but ESSL is better for cyclic transformations. Still, DUET outperforms the SimCLR baseline for cyclic transformations.

Interestingly, neither DUET nor ESSL outperform SimCLR by becoming equivariant to horizontal flips, as discussed in Section 6. Nevertheless, DUET still outperforms ESSL for horizontal flips by 0.86%, 4.4% and 4.18% on CIFAR-10, CIFAR-100 and TinyImageNet respectively. Similarly, as the dataset complexity increases, DUET performs better than ESSL for vertical flips. Another interesting result is the effectiveness of ESSL with rotations, where DUET remains subpar but better than the SimCLR baseline. The RRC column shows that DUET, by just learning structure to scale (approximately, as explained in Section 3.2), can improve accuracy using the vanilla SimCLR augmentation stack.

It is surprising how well DUET$_{\lambda=0}$ performs in the full stack setting, surpassing DUET for simpler datasets. Indeed, DUET$_{\lambda=0}$ learns an unsupervised structure, thus accounting for the interdependencies between the transformations applied. However, as observations of the transformation of interest are scarcer (*e.g.,* more complex datasets or less data per class) optimizing for a known structure is beneficial.

5.4. Transfer to Other Datasets

DUET's structure to rotations yields a gain of +21% with respect to SimCLR when transferring to Caltech101 (Li et al., 2022), and between +5.97% and +16.97% when transferring to other datasets like CIFAR-10, CIFAR-100, DTD (Cimpoi et al., 2014) or Oxford Pets (Parkhi et al., 2012). Structure to color transformations also proves beneficial, with a +6.36% gain on Flowers (Tung, 2020) (grayscale), Food101 (Bossard et al., 2014) (hue) and +7.13% on CIFAR-100 (hue). Horizontal flip is the transformation that sees less gain due to its ambiguity, as discussed in Section 6.

Table 2. RRC+1 results: Accuracy of a linear tracking head on CIFAR-100 and TinyImageNet. We also show the average over cyclic (vM target) and non-cyclic ($\mathcal{N}$ target) transformations. DUET improves over SimCLR for all groups, while ESSL worsens performance for color transformations. We report the mean_{std} over 3 runs.

Dataset	Method	RRC	Rot. (360)	Rot. (4-fold)	H. Flip	V. Flip	Avg.	Grayscale	Brightness	Contrast	Saturation	Hue	Avg.
CIFAR-100	SimCLR	$38.89_{0.26}$	$32.17_{0.14}$	$35.52_{0.33}$	$39.85_{0.13}$	$36.68_{0.27}$	$36.06_{0.18}$	$47.41_{0.40}$	$45.00_{0.26}$	$44.51_{0.04}$	$42.27_{0.70}$	$43.61_{0.19}$	$44.56_{0.26}$
	ESSL	-	$38.03_{0.36}$	$44.36_{0.72}$	$38.78_{0.54}$	$42.17_{0.87}$	$40.84_{0.51}$	$34.20_{0.60}$	$38.33_{0.25}$	$38.66_{0.02}$	$38.92_{0.08}$	$37.64_{0.39}$	$37.55_{0.22}$
	$\text{DUET}_{\lambda=0}$	$42.63_{0.11}$	$34.77_{0.72}$	$38.28_{0.19}$	$\mathbf{43.54}_{0.67}$	$40.10_{0.82}$	$39.17_{0.49}$	$48.87_{0.18}$	$48.12_{0.19}$	$47.74_{0.57}$	$45.39_{0.34}$	$46.32_{0.61}$	$47.29_{0.31}$
	DUET	$\mathbf{45.25}_{0.10}$	$\mathbf{42.17}_{0.42}$	$\mathbf{47.25}_{0.26}$	$41.82_{0.46}$	$\mathbf{45.38}_{0.73}$	$\mathbf{44.16}_{0.38}$	$\mathbf{50.91}_{0.49}$	$\mathbf{50.18}_{0.45}$	$\mathbf{49.77}_{0.46}$	$\mathbf{48.75}_{0.22}$	$\mathbf{48.54}_{0.78}$	$\mathbf{49.63}_{0.39}$
TinyImageNet	SimCLR	$26.91_{0.13}$	$21.34_{0.08}$	$24.40_{0.24}$	$27.90_{0.37}$	$26.74_{0.30}$	$25.09_{0.20}$	$31.35_{1.29}$	$29.95_{0.14}$	$29.68_{0.18}$	$28.60_{0.29}$	$28.20_{0.40}$	$29.55_{0.38}$
	ESSL	-	$25.35_{0.10}$	$30.41_{0.25}$	$27.13_{0.14}$	$29.11_{0.27}$	$28.00_{0.16}$	$23.51_{0.18}$	$26.45_{0.07}$	$26.32_{0.61}$	$26.75_{0.72}$	$26.00_{0.11}$	$25.80_{0.28}$
	$\text{DUET}_{\lambda=0}$	$29.57_{0.47}$	$24.22_{0.30}$	$26.96_{0.37}$	$\mathbf{30.78}_{0.23}$	$28.98_{0.42}$	$27.73_{0.27}$	$31.55_{0.14}$	$32.54_{0.18}$	$32.23_{0.17}$	$31.43_{0.24}$	$30.45_{0.60}$	$31.64_{0.22}$
	DUET	$\mathbf{31.26}_{0.21}$	$\mathbf{27.78}_{0.24}$	$\mathbf{31.55}_{0.28}$	$30.34_{0.59}$	$\mathbf{31.71}_{0.53}$	$\mathbf{30.34}_{0.33}$	$\mathbf{34.92}_{0.24}$	$\mathbf{33.96}_{0.16}$	$\mathbf{34.20}_{0.34}$	$\mathbf{33.42}_{0.35}$	$\mathbf{32.64}_{0.03}$	$\mathbf{33.83}_{0.18}$

Table 3. Full Stack results. We show the average accuracy of a linear tracking head over cyclic (vM target) and non-cyclic ($\mathcal{N}$ target) transformations. As the task complexity increases, DUET achieves better accuracy than the compared methods. Columns *Rot. (360)*, *Rot. (4-fold)* and *V. Flip* require an additional transformation. We report the mean_{std} over 3 runs.

Dataset	Method	RRC	Rot. (360)	Rot. (4-fold)	H. Flip	V. Flip	Avg.	Grayscale	Brightness	Contrast	Saturation	Hue	Avg.
CIFAR-10	SimCLR	$87.42_{0.01}$	$79.90_{0.50}$	$81.50_{0.44}$	$87.48_{0.06}$	$82.78_{0.23}$	$82.92_{0.22}$	$87.41_{0.03}$	$\mathbf{87.51}_{0.11}$	$\mathbf{87.57}_{0.19}$	$87.49_{0.08}$	$87.67_{0.34}$	$\mathbf{87.53}_{0.11}$
	ESSL	-	$\mathbf{86.55}_{0.13}$	$\mathbf{89.33}_{0.32}$	$84.78_{0.40}$	$\mathbf{86.66}_{0.21}$	$\mathbf{86.83}_{0.22}$	$83.59_{0.43}$	$85.78_{0.25}$	$86.31_{0.16}$	$87.12_{0.30}$	$86.39_{0.44}$	$85.84_{0.26}$
	$\text{DUET}_{\lambda=0}$	$\mathbf{87.50}_{0.20}$	$79.05_{0.37}$	$81.32_{0.18}$	$\mathbf{87.73}_{0.19}$	$82.66_{0.14}$	$82.69_{0.22}$	$\mathbf{87.69}_{0.17}$	$87.47_{0.13}$	$87.34_{0.20}$	$\mathbf{87.54}_{0.33}$	$87.63_{0.20}$	$\mathbf{87.53}_{0.20}$
	DUET	$87.22_{0.10}$	$81.70_{0.30}$	$83.49_{0.16}$	$85.64_{0.08}$	$83.84_{0.22}$	$83.67_{0.15}$	$87.40_{0.08}$	$86.97_{0.22}$	$87.05_{0.37}$	$87.52_{0.19}$	$\mathbf{87.97}_{0.11}$	$87.38_{0.16}$
CIFAR-100	SimCLR	$61.40_{0.17}$	$56.40_{0.30}$	$57.32_{0.03}$	$61.48_{0.28}$	$56.73_{0.48}$	$57.98_{0.19}$	$61.43_{0.21}$	$61.31_{0.05}$	$61.68_{0.57}$	$61.57_{0.41}$	$61.30_{0.03}$	$61.46_{0.18}$
	ESSL	-	$\mathbf{58.32}_{0.06}$	$\mathbf{63.28}_{0.28}$	$55.22_{0.29}$	$57.18_{0.18}$	$\mathbf{58.50}_{0.17}$	$55.10_{0.47}$	$57.92_{0.38}$	$58.06_{0.58}$	$60.25_{0.34}$	$58.91_{0.30}$	$58.05_{0.34}$
	$\text{DUET}_{\lambda=0}$	$62.13_{0.13}$	$55.49_{0.22}$	$57.79_{0.40}$	$\mathbf{62.25}_{0.34}$	$56.88_{0.18}$	$58.10_{0.29}$	$\mathbf{62.32}_{0.26}$	$\mathbf{62.39}_{0.27}$	$\mathbf{62.47}_{0.16}$	$62.54_{0.20}$	$62.29_{0.29}$	$62.40_{0.24}$
	DUET	$\mathbf{62.17}_{0.28}$	$55.66_{0.39}$	$58.01_{0.31}$	$59.62_{0.11}$	$\mathbf{57.40}_{0.15}$	$57.67_{0.20}$	$62.18_{0.51}$	$62.24_{0.31}$	$61.90_{0.72}$	$\mathbf{62.67}_{0.19}$	$\mathbf{63.31}_{0.21}$	$\mathbf{62.46}_{0.32}$
TinyImageNet	SimCLR	$42.16_{0.16}$	$37.35_{0.19}$	$39.23_{0.15}$	$42.31_{0.06}$	$39.35_{0.09}$	$38.50_{0.09}$	$42.11_{0.23}$	$42.32_{0.08}$	$42.34_{0.10}$	$42.27_{0.01}$	$42.46_{0.27}$	$42.30_{0.10}$
	ESSL	-	$37.53_{0.21}$	$\mathbf{42.86}_{0.29}$	$36.25_{0.13}$	$37.18_{0.77}$	$38.46_{0.29}$	$35.50_{0.30}$	$37.94_{0.13}$	$38.66_{0.50}$	$40.55_{0.74}$	$40.49_{0.05}$	$38.63_{0.28}$
	$\text{DUET}_{\lambda=0}$	$43.07_{0.11}$	$36.28_{0.95}$	$39.54_{0.41}$	$\mathbf{42.43}_{0.33}$	$38.87_{0.40}$	$39.28_{0.52}$	$\mathbf{42.79}_{0.16}$	$42.61_{0.34}$	$42.98_{0.08}$	$42.86_{0.30}$	$42.90_{0.46}$	$42.83_{0.27}$
	DUET	$\mathbf{43.56}_{0.54}$	$\mathbf{38.06}_{0.06}$	$40.03_{0.28}$	$40.43_{0.21}$	$\mathbf{39.36}_{0.50}$	$\mathbf{39.47}_{0.18}$	$42.55_{1.27}$	$\mathbf{43.41}_{0.01}$	$\mathbf{43.71}_{0.07}$	$\mathbf{44.13}_{0.57}$	$\mathbf{44.61}_{0.10}$	$\mathbf{43.68}_{0.29}$

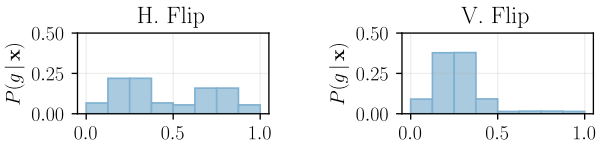
6. Discussion and Limitations

On the Dimensionality of DUET Representations. We reshape the output of the backbone ($\mathbb{R}^D$) to $\boldsymbol{z} \in \mathbb{R}^{C \times G}$. The final representation used for downstream tasks is a flattened ($\mathbb{R}^D$) version of $\boldsymbol{z}$. For a fair comparison, SimCLR and ESSL also yield $\mathbb{R}^D$ representations.

Trading off Structure and Expressivity. By increasing G we reduce the effective dimensionality of the content representations ($\mathbb{R}^C$) contrasted through $L_{\mathbb{C}}$. This implies a trade-off between structure (improves generation, transferrability) and expressivity (improves discrimination). Such effect is visible in the transfer learning results, where learning structure to rotation is not useful when transferring to the Flowers dataset. Indeed, such dataset contains many circular flowers, which are rotation (and flip) invariant.

Transformation Ambiguity. A dataset containing examples related by input transformations results in *transformation ambiguity*, and the distribution over group actions $P(\mathrm{g}|\boldsymbol{x}_k)$ becomes multi-modal. This is shown in Figure 5 where the weight of each mode corresponds to the observed probability in the dataset, i.e., $P(\mathrm{g}|\boldsymbol{x}_k)$ reflects the bias of the dataset with respect to the transformation. Additional results in Appendix J show ambiguity also for color transformations, *e.g.,* natural images may present a different default hue, yielding a spread $P(\mathrm{g}|\boldsymbol{x}_k)$. This phenomenon is also observed in Section 5.3.2 and Section 5.4, where the notion of a left-flipped image is ambiguous, whereas a vertically flipped image is not, and only the latter transformation yielded a performance gap between equivariant and invariant methods.

Figure 5. Observed $P(\mathrm{g}|\boldsymbol{x})$ for horizontal (left) and vertical (right) flips, obtained from 1000 CIFAR-10 images. Note the inherent ambiguity for horizontal flips. Also, see that the modes of the distributions correspond to the mapped points specified in Table 1.



Are c and g Dependent? To better understand the dependency between c and g (conditioned on $\boldsymbol{x}_k$) quantitatively, we measure the difference $\Delta P = \|P(\mathrm{c}, \mathrm{g}|\boldsymbol{x}_k) - P(\mathrm{c}|\boldsymbol{x}_k)P(\mathrm{g}|\boldsymbol{x}_k)\|_2^2$. In Table 4 we report the average difference for the DUET representations of 100 images from CIFAR-10, for 100 images with independent and identically distributed (iid) pixels and for 100 random representations

Table 4. Dependence of c and g conditioned on $\boldsymbol{x}_k$. The learnt marginal representations for content (c) and group element (g) are dependent. This is a core strength of DUET, where group structure and content are not assumed independent, but rather with specific dependencies learnt from data.

	ΔP
DUET w/ CIFAR-10	178.17
DUET w/ iid pixels	0.015
iid representations	0.00075

(iid features). Note that such difference is expected to be 0 for the random representations (independent) and close to 0 for the iid pixels (no symmetries in the data).

Computational Requirements The training time of SimCLR and DUET are practically the same. DUET's extra requirements suppose a negligible overhead, namely: are a sum over rows and cols of $\boldsymbol{z}$ and the computation of the Jensen-Shannon Divergence in $L_{\mathbb{C}}$. Interestingly, the projection head h in DUET is smaller than in SimCLR, since the content features are of lower dimension, effectively reducing the model parameters with respect to SimCLR.

Compared to ESSL, DUET shows an important computational gain. Indeed, the time required for ESSL to train depends on the group chosen. Taking the implementation in (Dangovski et al., 2022) for 4-fold rotations, the backbone consumes $2 + 4$ versions of each image, resulting in an overall training time $2.01\times$ longer than that of DUET. For other transformations, ESSL requires $2 + 2$ images being consumed (*e.g.,* flips) or $2 + 1$ (*e.g.,* contrast); thus resulting in longer training time than DUET in all cases.

7. Conclusion

We introduce DUET, a method to learn structured and equivariant representations using MSSL. DUET uses 2d representations that model the joint distribution between input content and the group element acting on the input. DUET representations, optimized through the content and group element marginal distributions, become structured and equivariant to the group elements. We design an explicit form of transformation at representation level that allows exploiting equivariance for controlled generation. Our results show that DUET representations are expressive for generative purposes (lower reconstruction error) and also for discriminative purposes. Overall, this work shows that accounting for the topological structure of input transformations is of great importance to improve generalization in MSSL.

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *NeurIPS*, volume 32, 2019.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *ICML*, pp. 2990–2999. PMLR, 2016.

Cotogni, M. and Cusano, C. Offset equivariant networks and their applications. *Neurocomputing*, 502:110–119, 2022.

Dangovski, R., Jing, L., Loh, C., Han, S., Srivastava, A., Cheung, B., Agrawal, P., and Soljačić, M. Equivariant contrastive learning. *ICLR*, 2022.

Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *CVPR*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *ICLR*, 2019.

Huang, C., Goh, H., Gu, J., and Susskind, J. Mast: Masked augmentation subspace training for generalizable self-supervised priors. In *ICLR*, pp. 297–304, 2023.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37, pp. 448–456, 2015.

Jiao, J. and Henriques, J. F. Quantised transforming auto-encoders: Achieving equivariance to arbitrary transformations in deep networks. In *BMVC*, 2021.

Keller, T. A. and Welling, M. Topographic VAEs learn equivariant capsules. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Keller, T. A., Suau, X., and Zappella, L. Homomorphic self-supervised learning. *NeurIPS SSL Workshop*, 2022.

Kim, S., Kim, S., and Lee, J. Hybrid generative-contrastive representation learning. *ICLR*, 2021.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. pp. 32–33, 2009.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. In *ICLR*, 2018.

Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks, 2016.

Lee, H., Lee, K., Lee, K., Lee, H., and Shin, J. Improving transferability of representations via augmentation-aware self-supervision. *NeurIPS*, 2021.

Li, F.-F., Karpathy, A., and Johnson, J. cs231n course at stanford university, 2017. URL https://www.kaggle.com/c/tiny-imagenet.

Li, F.-F., Andreeto, M., Ranzato, M., and Perona, P. Caltech 101, 2022.

Li, T., Fan, L., Yuan, Y., He, H., Tian, Y., Feris, R., Indyk, P., and Katabi, D. Addressing feature suppression in unsupervised visual representations. *arXiv preprint arXiv:2012.09962*, 2020.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Löwe, S., O'Connor, P., and Veeling, B. Putting an end to end-to-end: Gradient-isolated learning of representations. In *NeurIPS*, 2019.

MacDonald, L. E., Ramasinghe, S., and Lucey, S. Enabling equivariance for arbitrary lie groups. pp. 8183–8192, 2022.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *NeurIPS*, 2018.

Pal, D. K. and Savvides, M. Non-parametric transformation networks, 2018.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *CVPR*, 2012.

Serre, J.-P. *Linear representations of finite groups.*, volume 42 of *Graduate texts in mathematics*. Springer, 1977.

Sosnovik, I., Szmaja, M., and Smeulders, A. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020.

Stühmer, J., Turner, R., and Nowozin, S. Independent subspace analysis for unsupervised learning of disentangled representations. In *AISTATS*, 2020.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *NeurIPS*, 2020.

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. On mutual information maximization for representation learning, 2019.

Tung, K. Flowers Dataset, 2020. URL https://doi.org/10.7910/DVN/1ECTVN.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, volume 119, pp. 9929–9939. PMLR, 2020.

Wu, Y. and He, K. Group normalization. In *ECCV*, pp. 3–19, 2018.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. *ICML*, 2021.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. Contrastive learning inverts the data generating process. In Meila, M. and Zhang, T. (eds.), *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12979–12990. PMLR, 2021.

A. Bounding the Equivariance Error

To introduce some notation, let us assume that, for an input data point $\boldsymbol{x}_0$, the training procedure has "seen" the augmentations $\boldsymbol{x}_i = \tau_{g_i}(\boldsymbol{x}_0)$, generating the respective representations $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ in feature space. Notice that, with some abuse of notation , in this scenario we consider $\boldsymbol{z}$ to be the column reduction of the feature space, since that is the only part dedicated to guaranteeing equivariance. Since we are at the optimum, these must produce group marginal distributions $Q_j(z_i) \equiv \hat{Q}_{g_i}$, where $\hat{Q}_{g_i}$ represents the discretization of the target distribution with mean g_i. At the end of training, if exact equivariance is reached (i.e. if $L_{\mathbb{G}}$ is minimized), a newly generated augmentation $\boldsymbol{x} = \tau_g(\boldsymbol{x}_0)$ for a given transformation parameter g, would be mapped by our neural network to the feature vector $\boldsymbol{z}$, such that $Q(\boldsymbol{z}) \equiv \hat{Q}_g$. Since this augmentation was not seen during training time, however, this is not guaranteed. We are interested in providing a bound on the error between the representation actually recovered, and the ideal one, which gives us an indication of how much our neural network can violate equivariance for unseen transformation parameters g. This is given by the following theorem.

Theorem A.1. *For a training point $\boldsymbol{x}_0$, at the optimum of $L_{\mathbb{G}} = 0$, the equivariance error of a neural network f trained with loss Equation (2) is bounded by*

$$\|f(\tau_g(\boldsymbol{x}_0)) - T_g(f(\boldsymbol{x}_0))\| \leq (L_f L_{\tau_g} + L_{T_g}) \min_i |g - g_i|, \tag{7}$$

where L_{T_g}, L_{τ_g} and L_f are the Lipschitz constants associated with the transformations T_g, τ_g, and the network f, respectively.

Proof. Using triangular inequality, we get

$$\|f(\tau_g(\boldsymbol{x}_0)) - T_g(f(\boldsymbol{x}_0))\| \leq \|f(\tau_g(\boldsymbol{x}_0)) - f(\tau_{g_i}(\boldsymbol{x}_0))\| + \|f(\tau_{g_i}(\boldsymbol{x}_0)) - T_g(f(\boldsymbol{x}_0))\| \tag{8}$$

for any given augmentation $\boldsymbol{x}_i = \tau_{g_i}(\boldsymbol{x}_0)$ seen during training time. At the optimum, we have by construction that $f(\tau_{g_i}(\boldsymbol{x}_0)) = T_{g_i}(f(\boldsymbol{x}_0))$, which allows us to rewrite the second term as

$$\|f(\tau_{g_i}(\boldsymbol{x}_0)) - T_g(f(\boldsymbol{x}_0))\| = \|T_{g_i}(f(\boldsymbol{x}_0)) - T_g(f(\boldsymbol{x}_0))\| \leq L_{T_g}|g - g_i| \tag{9}$$

Notice L_{T_g} depends on the target discretization chosen: for the Gaussian target and ∞-norm, we recover it analytically in Lemma A.3. The first term, instead, becomes

$$\|f(\tau_g(\boldsymbol{x}_0)) - f(\tau_{g_i}(\boldsymbol{x}_0))\| \leq L_f \|\tau_g(\boldsymbol{x}_0) - \tau_{g_i}(\boldsymbol{x}_0)\| \leq L_f L_{\tau_g}|g - g_i|. \tag{10}$$

Combining these results together, we recover the target bound. □

Notice that for a *discrete* group, instead, it is possible to train $f(\boldsymbol{x})$ so that it achieves exact equivariance:

Corollary A.2. *Given a discrete group $\mathcal{G}$, a neural network $f(\boldsymbol{x})$ trained with loss Equation (2) achieves equivariance at the optimum, if it is exposed to all group transformations.*

Proof. The proof follows directly from Theorem A.1 by noticing that $g - g_i = 0$ necessarily, if all group transformations have been seen during training time. □

Theorem A.3. *For a Gaussian target, $\hat{Q}_i(g) = \frac{\int_{\Omega_i} \mathcal{N}(g,\sigma)(\tilde{g})\, d\tilde{g}}{\int_{[0,1]} \mathcal{N}(g,\sigma)(\tilde{g})\, d\tilde{g}}$, the Lipschitz continuity constant for T_g in ∞-norm is given by $\hat{\mu}'_{G-1}(0)$, with $\hat{\mu}_j$ defined in Equation (5).*

Proof. Starting from the definition of $T_g(\boldsymbol{z})$ in Equation (6), and using the mean-value theorem, we get

$$\|T_g(\boldsymbol{z}) - T_{\hat{g}}(\boldsymbol{z})\|_\infty = \|\hat{\boldsymbol{M}}_g - \hat{\boldsymbol{M}}_{\hat{g}}\|_\infty = \max_j |\hat{\mu}_j(g) - \hat{\mu}_j(\hat{g})| = \max_j |\hat{\mu}'_j(\tilde{g}_j)||g - \hat{g}| \tag{11}$$

for some (possibly different for different j) $\tilde{g}_j \in [g, \hat{g}]$. We remind that $\hat{\mu}_j(g)$ is defined in equation 5 as

$$\begin{aligned}\hat{\mu}_j(g) &= \ln \hat{Q}_j(g) - \frac{1}{G}\sum_i \ln \hat{Q}_i(g) = \ln\left(\frac{\Delta\Phi_j^{j+1}(g)}{\Delta\Phi_0^G(g)}\right) - \frac{1}{G}\sum_i \ln\left(\frac{\Delta\Phi_i^{i+1}(g)}{\Delta\Phi_0^G(g)}\right) \\ &= \ln \Delta\Phi_j^{j+1}(g) - \frac{1}{G}\sum_i \ln \Delta\Phi_i^{i+1}(g), \quad \text{with} \quad \Delta\Phi_i^j = \int_{g_i}^{g_j} \mathcal{N}(g,\sigma)(x)\, dx, \quad \text{and} \quad g_i = \frac{i}{G},\end{aligned} \tag{12}$$

so that its derivative can be compactly written as

$$\hat{\mu}_j'(g) = h_j(g) - \frac{1}{G}\sum_i h_i(g), \quad \text{where} \quad h_i(g) = \frac{(\Delta\Phi_i^{i+1})'(g)}{\Delta\Phi_i^{i+1}(g)}. \tag{13}$$

Our goal is to bound $\hat{\mu}_j'(g)$, which can be quantified starting from considerations on the various $h_j(g)$. It can be proven that these are:

- equivalent modulo translations: $h_j(g) = h_{j-i}(g - i/G)$;
- antisymmetric with respect to g around the centerpoint $g_j^* = (g_{j+1} + g_j)/2$: $h_j(g_j^* + g) = -h_j(g_j^* - g)$;
- antisymmetric with respect to j: $h_j(g_j^* + g) = -h_{G-j}(g_{G-j}^* - g)$;
- decreasing: $h_j'(g) \leq 0$;
- convex for $g < g_j^*$: $g \lesseqgtr g_j^* \implies h_j''(g) \gtreqless 0$.

We can gain a better intuition about how to effectively bound $\hat{\mu}'(g)$ by rewriting equation 13 using the equivalence under translations of $h_j(g)$:

$$\hat{\mu}_j'(g) = \frac{1}{G}\sum_i \left(h_j(g) - h_j\left(g + \frac{j-i}{G}\right)\right) \tag{14}$$

this shows that for each j we are averaging the differences between $h_j(g)$ and the same function evaluated at G equispaced points $g - (i-j)/G$. Since $h_j(g)$ is decreasing, we deduce that this difference is positive whenever $i < j$, and negative otherwise. We have then that the maximum absolute value of $\hat{\mu}_j'(g)$ is always attained for the most extreme j, since that guarantees that the largest number of terms share the same sign. Without loss of generality (by symmetry), we can consider $j = G-1$, and we have

$$\max_j |\hat{\mu}_j'(g)| = \hat{\mu}_{G-1}'(g) \qquad \forall g \in [0,1]. \tag{15}$$

It suffices now to bound this quantity in $[0,1]$. Due to the concavity of $h_j(g)$, its maxima will be at the boundary, and specifically at $g = 0$. This can be shown by simply comparing the values at 0 and at 1 (we drop the subscript $G-1$ and consider $h_{G-1}(g) = h(g)$ from now on):

$$\begin{aligned}
\hat{\mu}_{G-1}'(0) - \hat{\mu}_{G-1}'(1) &= \frac{1}{G}\sum_{i=0}^{G-1}\left(h(0) - h\left(\frac{G-1-i}{G}\right)\right) - \frac{1}{G}\sum_{i=0}^{G-1}\left(h(1) - h\left(1 + \frac{G-1-i}{G}\right)\right) \\
&= \frac{1}{G}\sum_{i=0}^{G-1}\left(h(0) + h\left(\frac{G-1}{G}\right) - 2h\left(\frac{i}{G}\right)\right) \\
&= \frac{1}{G}\sum_{i=0}^{G-1}\left(h(0) + h\left(\frac{G-1}{G}\right) - \left(h\left(\frac{i}{G}\right) + h\left(\frac{G-1-i}{G}\right)\right)\right) \geq 0
\end{aligned} \tag{16}$$

where we exploited the antisymmetry of $h_{G-1}(g)$ around $g_{G-1}^* = 1 - 1/(2G)$ to aptly change the inner arguments, as well as the convexity of $h(g)$ for $g < g_{G-1}^*$ to state that $h(0) + h\left(\frac{G-1}{G}\right) \geq \left(h\left(\frac{i}{G}\right) + h\left(\frac{G-1-i}{G}\right)\right)$, for each i. This allows us to explicitly write the Lipschitz constant for the transformation $T_g(\boldsymbol{z})$ as

$$\|T_g(\boldsymbol{z}) - T_{\hat{g}}(\boldsymbol{z})\|_\infty \leq \hat{\mu}_{G-1}'(0)|g - \hat{g}|. \tag{17}$$

□

B. Proof of Axioms for T_g in Equation (6)

Notice that T_g, thus defined, satisfies the group axioms at proper training ($L_{\mathbb{G}}$ is minimized). In fact:

- Neutral: $g = 0 \;\; s.t. \;\; T_0(\boldsymbol{z}) = \boldsymbol{z}$. Easily proven since $\boldsymbol{M}_{g_0} = \widehat{\boldsymbol{M}}_{g_0+0}$.

- Inverse: $g^{-1} = -g \;\; s.t. \;\; T_{g^{-1}} \circ T_g(\boldsymbol{z}) = \boldsymbol{z}$. Let $\boldsymbol{z}' = T_g(\boldsymbol{z})$, then $T_{g^{-1}}(\boldsymbol{z}') = \boldsymbol{z}' - \boldsymbol{M}_{g_0'} + \widehat{\boldsymbol{M}}_{g_0'-g}$. Since $g_0' = g_0 + g$, then $T_{g^{-1}} \circ T_g(\boldsymbol{z}) = \boldsymbol{z} - \boldsymbol{M}_{g_0} + \widehat{\boldsymbol{M}}_{g_0+g} - \boldsymbol{M}_{g_0+g} + \widehat{\boldsymbol{M}}_{g_0+g-g} = \boldsymbol{z}$.
- Associativity: Similar reasoning as for the inverse property with 2 different elements.
- Closure: We work in $\mathbb{R}^D$ at representation level, so closure is verified.

C. DUET for Multiple Groups

Modern MSSL frameworks use complex augmentation stacks that compose several transformations. While learning structure with respect to a single group is interesting, one could benefit from learning such structure for a set of groups. For readability, we focus on the two group case ($\mathcal{G}_A$ and $\mathcal{G}_B$); but the following reasoning can easily be extended to more groups.

In order to model the interdependencies between groups and content, one can learn the joint distribution $P(\mathrm{c}, \mathrm{g}_A, \mathrm{g}_B|\boldsymbol{x}_k)$, where $\mathrm{g}_A \in \mathbb{G}_A$ and $\mathrm{g}_B \in \mathbb{G}_B$ are 2 random variables representing the respective group elements. Such approach implies that our backbone f maps to $\mathbb{R}^{C\times|\mathbb{G}_A|\times|\mathbb{G}_B|}$. The marginal distributions are now obtained by summing over the non-desired dimensions (*e.g.,* over $\mathbb{C}$ and $\mathbb{G}_A$ to obtain $P(\mathrm{g}_B|\boldsymbol{x}_k)$). Using these new marginals, we define the multi-group loss as

$$L_{\text{Multi-}\mathbb{G}} = \frac{1}{2} \sum_{l=\{A,B\}} L_{\mathbb{G}_A} + L_{\mathbb{G}_B}. \tag{18}$$

However, as the number of groups increases, modelling the joint distribution becomes intractable. In practice, keeping C constant, the dimensionality of $\boldsymbol{z}$ increases in $O(G^n)$ with the number of groups.

To address scalability, we propose to relax the formulation and let our backbone f map into $\mathbb{R}^{C\times(|\mathbb{G}_A|+|\mathbb{G}_B|)}$, so that the dimensionality of $\boldsymbol{z}$ increases in $O(nG)$ with the number of groups. Using this relaxation we actually consider $\mathbb{G}_A$ and $\mathbb{G}_B$ independent, although their structure is learnt jointly during training. In practice, $\boldsymbol{z}$ is divided into two blocks, with $|\mathbb{G}_A|$ and $|\mathbb{G}_B|$ columns each. In this scenario, $P(\mathrm{g}_A|\boldsymbol{x}_k)$ is obtained by summing over columns of the $\mathbb{G}_A$ block, and $P(\mathrm{g}_B|\boldsymbol{x}_k)$ by summing over the columns of the $\mathbb{G}_B$ block. The content marginal $P(\mathrm{c}|\boldsymbol{x}_k)$ is obtained by concatenating the sum over the rows of each group block.

D. Recovering the Transformation Parameter for a Von-Mises Distribution

Let $\boldsymbol{x}_i$ be samples of a $\mathrm{vM}(\boldsymbol{x}|\mu, \kappa)$ with unknown parameters μ and κ. We want to recover the parameter μ, which corresponds to the group element that yields such vM prior. Let $\boldsymbol{r} = \sum_i \boldsymbol{x}_i$ be the baricenter of the samples with respect to the origin, then $\tilde{g} = \tilde{\mu} = \mathrm{angle}(\boldsymbol{r})$.

E. Empirical Equivariance: Additional Plots

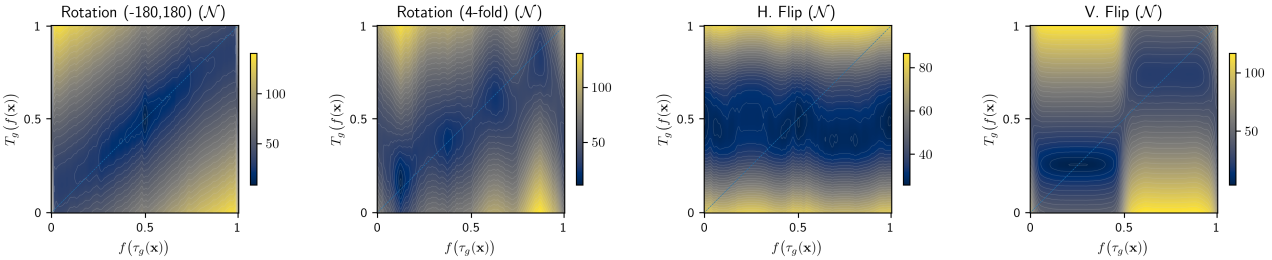
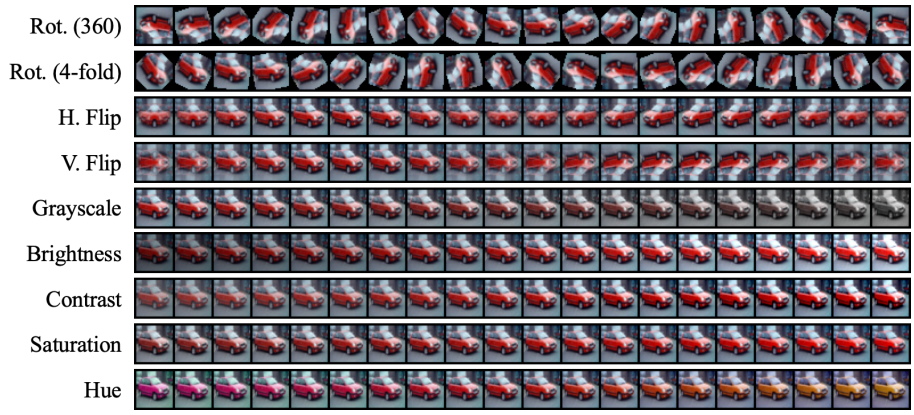


Figure 6. Empirical validation of equivariance for cyclic groups with a non-cyclic Gaussian prior. Note the difference with the top row of Figure 2. In the Gaussian case, the cyclic nature of rotation and flip is not observed, and equivariance is less well satisfied.

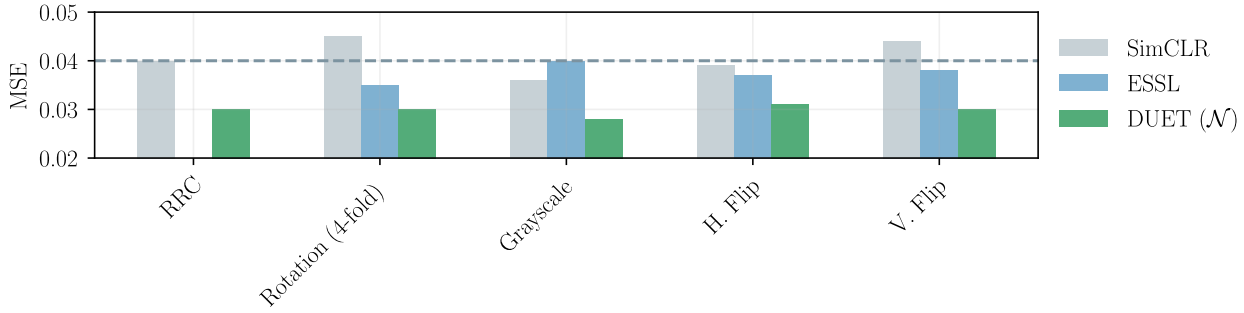
Figure 7. Examples of the transformations τ_g applied on the input images $\boldsymbol{x}$ to obtain the plots in Figure 2 and Figure 6. For flips, we simulate a gradual flip by alpha-blending the 2 flipped images.



F. Reconstruction Error

In order to verify our hypothesis that structured representations are beneficial for generation, we measure the reconstruction error obtained with the decoders used in Section 5.2. We use a mean squared error loss for reconstruction: $L_{rec}^{(i)} = \|d(f(\tau_g(\boldsymbol{x}^{(i)}))) - \tau_g(\boldsymbol{x}^{(i)})\|_2^2$, where $d(\cdot)$ is a decoder network. In Figure 8 we plot the final test L_{rec} on CIFAR-10 for decoders trained on frozen DUET, ESSL and SimCLR representations, for some of the transformations analyzed. The obtained reconstruction error with DUET is up to 66% smaller than with SimCLR (rotation (4-fold)) and up to 70% smaller than with ESSL (grayscale).

Figure 8. Reconstruction error (smaller is better) obtained with decoders trained on frozen DUET, ESSL and SimCLR representations. The horizontal dashed line shows the baseline error of SimCLR with only RRC.



G. Full Stack Augmentations

In Section 5.3.2 we report the performance of DUET and other methods using the full SimCLR augmentation stack. More precisely, the augmentations used are:

- `RandomResizedCrop(scale=(0.2, 1.0))`
- `ColorJitter(brightness=0.4, saturation=0.4, contrast=0.4, hue=0.1, p=0.8)`
- `RandomHorizontalFlip(p=0.5)`
- `RandomGrayscale(p=0.2)`
- `RandomGaussianBlur(kernel_size=(3, 3), sigma=(0.1, 2.0), p=0.5)`

When we learn structure for groups that are not directly parameterized in this stack, we add a specific transformation. For example, for the *vertical flip* group we add `RandomVerticalFlip(p=0.5)`. Or for rotations, we add a random rotation transformation in the stack.

H. Training Procedure

For all our experiments we use as backbone a ResNet-32 (He et al., 2016) architecture with an input kernel of 3×3 and stride of 1. The output dimensionality of the ResNet is $\mathbb{R}^{512}$, which we reshape to $\mathbb{R}^{64\times 8}$ for a group granularity of $G = 8$. Note that we do not add parameters, we only reshape the output of a vanilla ResNet to build our DUET representations. Additional training parameters are shown in Table 5.

The detached decoders in Section 5.2 are also trained using the same procedure. The reconstructed images are RGB with 32×32 pixels. The decoder architecture is a ResNet-18 with swish activation functions, visual attention and GroupNorm (Wu & He, 2018) normalization.

Table 5. Training parameters.

Batch size	2048
Epochs	800
Input images	RGB of 32×32
Learning rate	0.0001
Learning rate warm-up	10 epochs
Learning rate schedule	Cosine
Optimizer	Adam($\beta = [0.9, 0.95]$)
Weight decay	0.0001

H.1. Effect of λ, σ and G

We perform a sweeping of λ values between 0 and 1000. The first observation is that adding structure improves over SimCLR for all transformations (see Figure 10 in the Appendix). However, color transformations and horizontal flips degrade performance if we strongly impose structure. This result hints that structure for such groups is harder to learn, or is less learnable from data (*e.g.,* the structure is ambiguous, as in the case of having flipped and non-flipped images in the dataset). Interestingly, DUET$_{\lambda=0}$ also improves slightly over SimCLR, showing that unsupervised structure is still helpful for the specific case of CIFAR-10. Overall, our results show that $\lambda = 10$ is optimal for all transformations but *scale*, *rotations* and *vertical flips* which can handle up to $\lambda = 1000$.

In Figure 9 we show the accuracy of SimCLR and DUET at different σ for all transformations analyzed. The violin plots show the median accuracy across transformations. We obtain an empirically optimal value of $\sigma = 0.2$ for DUET. Note that $\sigma = 10$ is almost equivalent to a uniform target, thus not imposing any structure. In Figure 11 we show the detailed results per transformation, observing that horizontal flip behaves better with a uniform target. Indeed, as observed in Section 5.1 and Section 5.3, with the datasets used horizontal flip is ambiguous and we cannot learn this symmetry from data.

Lastly, we found DUET is quite insensitive to the choice of parameter G, based on results on CIFAR-10. We sweep $G = 2, 4, 8, 16$ and the obtained accuracy changes by less than 1%. We choose to use a reasonable value of $G = 8$.

Figure 9. Sweep of DUET's parameter σ. We find empirically that $\sigma \approx 0.2$ works best.

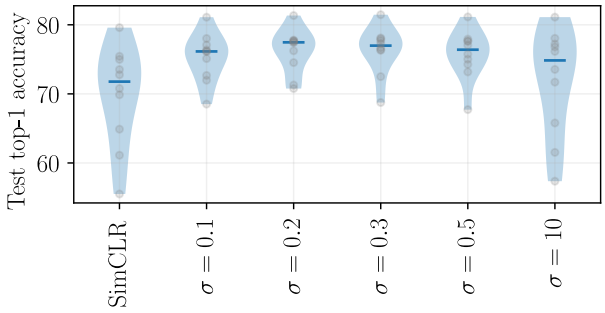


Figure 10. Sweep of DUET's parameter λ. Note that different transformations require different optimal λs.

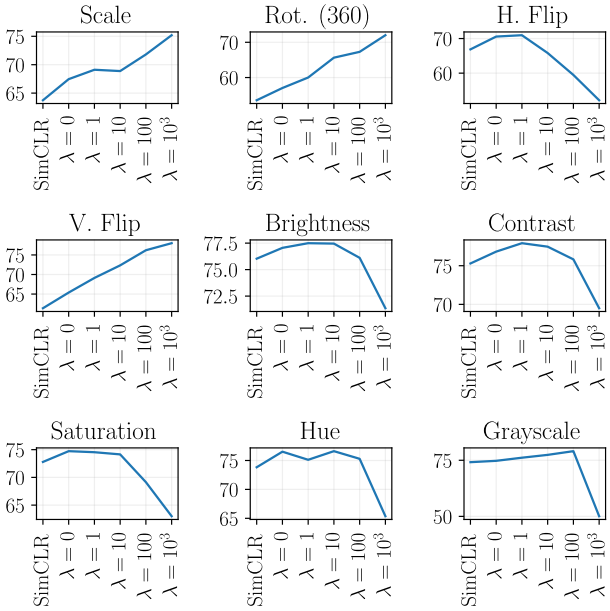
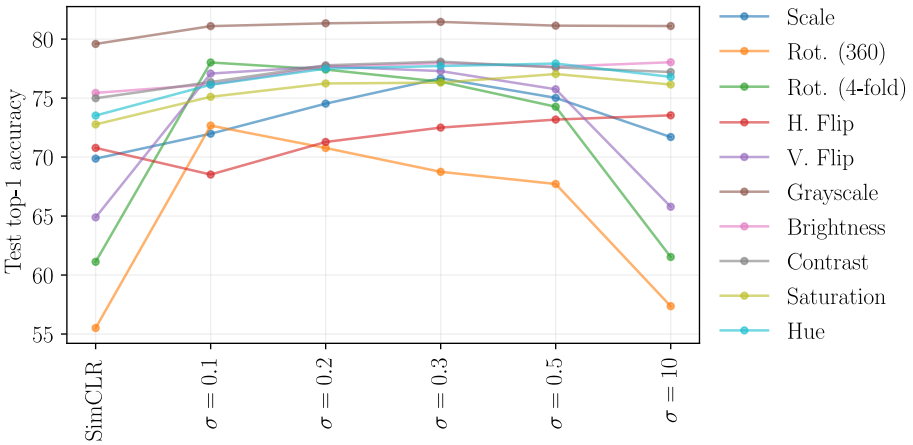


Figure 11. Test top-1 performance on CIFAR-10 as we modify the σ parameter in DUET. We report here the results per group.



I. Additional Results for Transfer Learning

These results complement those summarized in Section 5.4. We train a logistic regression classifier on the representations of the training split of each dataset. No augmentations are applied during the classifier training. At test time, we evaluate the classifier on the test set of each dataset.

In Figure 12 we report the difference in accuracies between DUET and SimCLR. DUET's structure to rotations yields a gain of +21% when transferring to Caltech101, and very important gains when transferring to other datasets like CIFAR-10, CIFAR-100, DTD or Pets. Structure to color transformations also proves beneficial, with a +6.36% gain on Flowers (grayscale), 7.05% on Food101 (Bossard et al., 2014) (hue) and 7.13% on CIFAR-100 (hue). Horizontal flip is the transformation that sees less gain, as expected given its ambiguity as shown in Figure 5.

It is interesting to see that DUET achieves slightly worse performance for rotations or flips on the Flowers dataset. Indeed, this dataset contains many *circular* flowers, which are rotation (or flip) invariant. In such situation, learning structure for rotations (or flips) should not give any gain. Actually, in DUET we are trading off content for structure, so if the structure learnt is not useful, we are actually diminishing the expressivity of the final representations.

Comparing with ESSL Figure 13, DUET achieves better transfer results for most of the datasets and transformations. Interestingly, ESSL improves over DUET for geometric transformations on Flowers, due to the trade-off inherent in DUET (see Section 6). For completeness, the results of ESSL compared to SimCLR are shown in Figure 14.

The linear regression for ESSL with grayscale on CIFAR-100 did not converge, thus we removed that result from the plots.

Figure 12. Difference in accuracies between DUET and SimCLR, when transferring representations learnt on TinyImageNet to different datasets in the RRC+1 setting.

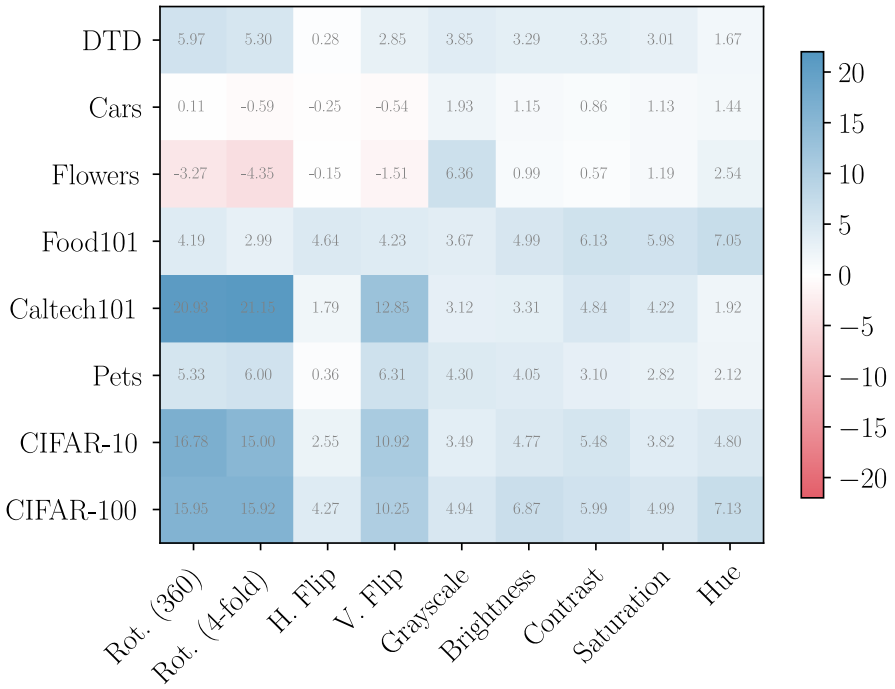


Figure 13. Difference in transfer accuracies between DUET and ESSL, when transferring representations learnt on TinyImageNet to different datasets in the RRC+1 setting.

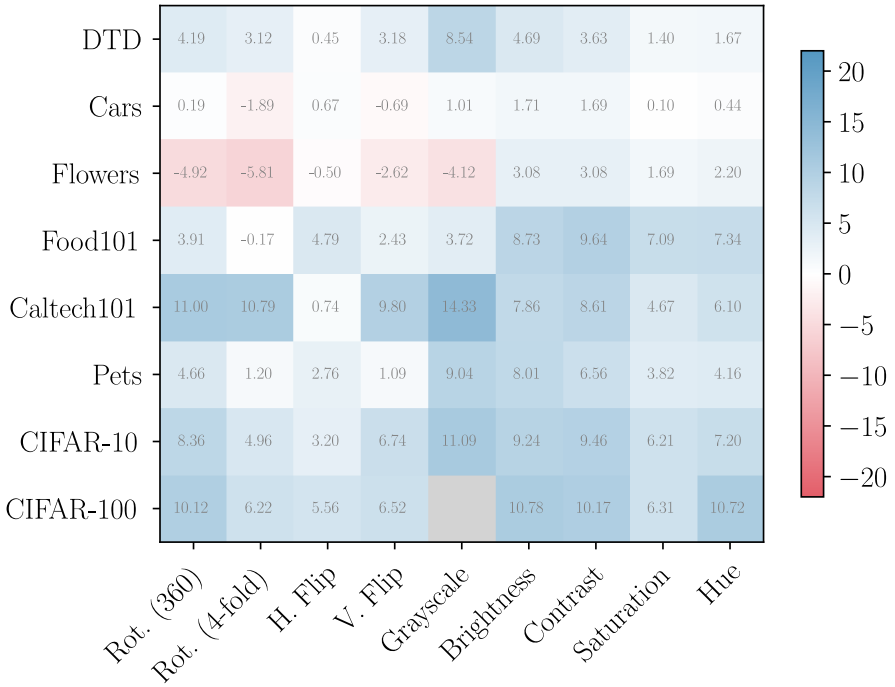
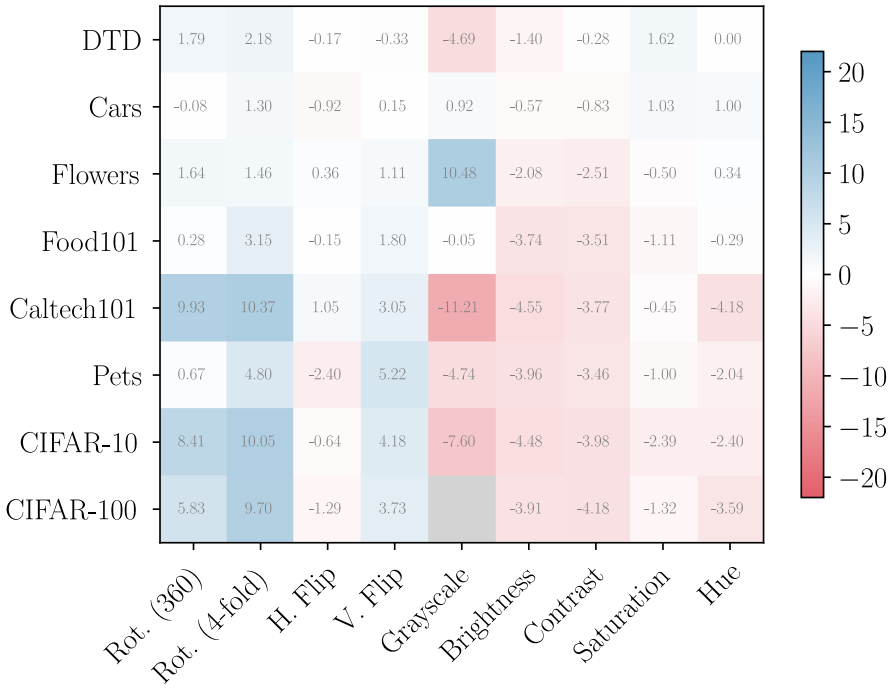


Figure 14. Difference in transfer accuracies between ESSL and SimCLR, when transferring representations learnt on TinyImageNet to different datasets in the RRC+1 setting.



J. Additional Results about Transformation Ambiguity

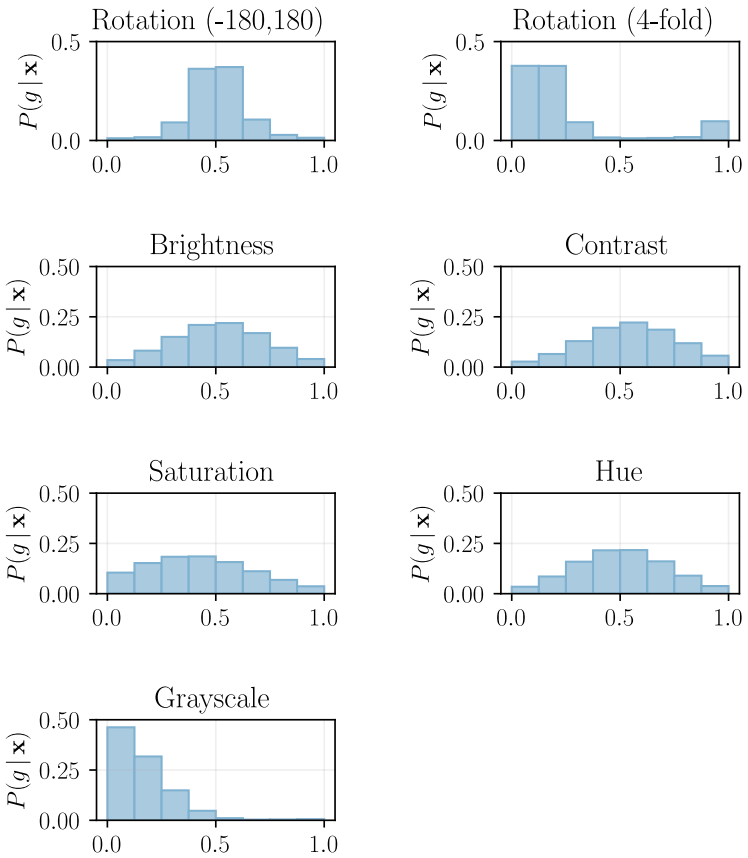



Figure 15. Observed $P(\mathrm{g}|\boldsymbol{x})$ for different transformations, obtained from 100 randomly sampled CIFAR-10 images. Note the inherent ambiguity for color transformations, in addition to the one observed for horizontal flips in Figure 5.(left). Also, see how the modes of the distributions correspond to the mapped points in Table 1.