A RELATIONSHIP BETWEEN RELIC AND OTHER METHODS

Table 5: The objective in eq. (3) recovers state of the art methods depending on design choices ("-" denotes the identity function and "norml." means g is constrained to have unit norm).

Method	ϕ	g	Regl.
CPC (Hénaff et al., 2019)	$\langle g, Wg \rangle$	PixelCNN	-
AMDIM (Bachman et al., 2019)	$\langle \cdot, \cdot \rangle$	-	-
SimCLR (Chen et al., 2020a)	$\langle g, g \rangle$	MLP, norml.	-
BYOL (Grill et al., 2020)	-	g_1, g_2 1 layer MLP, norml.	$\|g_1(g_2) - g_2\|^2$
RELIC (ours)	$\langle g, g \rangle$	MLP, norml.	Eq. (3)

B DISTANCE CONCENTRATION AND GENERALIZATION

Quantifying the generalization performance of representations learned on unlabelled data is a difficult task without imposing assumptions on the underlying structure of the data and the downstream tasks of interest. The results in (Saunshi et al., 2019) assume a latent class structure underlying the data. The similarity of images under each (potentially overlapping) latent class c is measured by a probability distribution $\mathcal{D}_c$. In the contrastive setting a positive pair of points $\{x, x^+\}$ is said to be sampled from a distribution $\mathbb{E}_c \mathcal{D}_c(x)\mathcal{D}_c(x^+)$ and a negative example x^- is sampled from the marginal distribution. The task of interest is multi-class classification using the learned representation. In our setting the augmented data points $\{x_i^{a_l}, x_i^{a_k}\}$ and $\{x_i^{a_l}, x_m^{a_k}\}_{m=1}^M$ take the roles of the pairs of positive and negative points, respectively.

In this section, under the same structural assumptions on the data as (Saunshi et al., 2019) we will show that a similar result holds but under weaker assumptions on the function, f.

To intuit the following results, we can view our explicit invariance constraint through the lens of distance concentration. Its effect can be seen intuitively in Figure 3. The shaded region represents the set of augmentations, $\mathcal{A}$ around an image. Depicted are two images x_i and x_j from the ImageNet class Stingray. The points $x_i^{a_l}$ and $x_j^{a_k}$ are augmentations which correspond to a region of overlap between the augmentation sets of x_i and x_j. If the augmentations $f(x_i^{a_l})$ and $f(x_j^{a_k})$ are similar enough, encouraging $f(x_i)$ to be close to $f(x_i^{a_l})$ and similarly for $f(x_j)$ and $f(x_j^{a_k})$ indirectly encourages $f(x_i)$ to be close to $f(x_j)$. This has the effect of concentrating distances between similar images. We will make this intuition more formal in the following discussion.

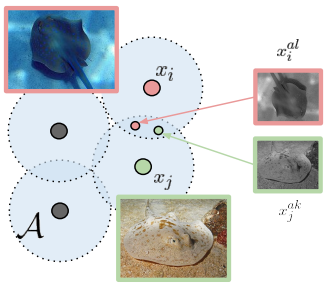

Figure 3: Visual representation of invariance penalty. Shaded region denotes set of augmentations around an image.

Consider a modified, Euclidean distance regularized version of our objective

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \sum_{a_l a_k} \ell(\{f(x_i^{a_l})^\top (f(x_i^{a_k}) - f(x_m^{a_k}))\}_{m=1}^M) \quad (5)$$
$$s.t. \quad \|f(x_i) - f(x_i^{ak})\|^2 \leq \rho.$$

where $f \in \mathcal{F} = \{f : \mathcal{X} \mapsto \mathbb{R}^d \; s.t. \; ||f||_2 \leq T\}$ with $T \geq 0$. Here $\ell(v) = \log(1 + \sum_m \exp(v_m))$ is the logistic loss. For a single negative, this is equivalent to the standard RELIC objective with an identity critic.

Assumption 1. *We require that the following assumptions hold:* **(A1)** *$\hat{f}$ is L-Lipschitz and minimizes eq. (5) such that the constraint is active and* **(A2)** *x is a bounded variable.*

Lemma 1 (Concentration)**.** *If assumption (A1) holds for $\rho \leq \frac{B}{6L\kappa}$, and (A2) holds for x, $\hat{f}(x)$ is a sub-Gaussian random variable with parameter $\sigma_f^2 \leq \frac{1}{\kappa}\sigma_x^2$.*

See Appendix C for proof. This result states that the Euclidean version of our invariance regularizer has the effect of contracting the within-class variance of the data. Figure 2 shows that this holds in practise for the original version of our objective in eq. (3). This guarantees that the following generalization result from (Saunshi et al., 2019) holds. For brevity we state an informal version of the Theorem with details deferred to the original publication.

Theorem 2 (Generalization. Adapted from Lemma B.2. from (Saunshi et al., 2019))**.** *Let $L^{\mu}_{sup}(f)$ be the standard $(K+1)$-wise hinge loss of the linear classification function $W^{\mu}f$ whose c^{th} column is $\mu_x = \frac{1}{|\mathcal{C}_c|}\sum_{i\in\mathcal{C}_c} f(x_i)$ the mean of representations corresponding to class c. Further, let $L^{\mu}_{\gamma(f),sup}(f)$ use the the hinge loss with margin $\gamma(f) = 1 + c'M\sigma_f(\sqrt{k} + \sqrt{\log\frac{M}{\epsilon}})$ with c' constant and $M = \max_x \|f(x)\|$. If $\hat{f}$ is the minimizer of eq. (5) and if Assumptions **(A1)** and **(A2)** hold then with high probability*

$$L^{\mu}_{sup}(\hat{f}) \leq \gamma(f)L^{\mu}_{\gamma(f),sup}(f) + Gen_N + \epsilon \tag{6}$$

Here, Gen_N is a standard generalization bound which depends on the Rademacher complexity of the function class $\mathcal{F}$ and the sample size, N.

For all practical purposes, the final generalization result is identical to (Saunshi et al., 2019) stating that $\hat{f}$—which is learned by minimizing a contrastive objective on unlabelled data—performs well on labelled data. However, this crucially depends on the intraclass concentration of the representation, that $f(x)$ is sub-Gaussian with parameter σ_f^2. Whereas in (Saunshi et al., 2019) this was assumed to hold, our Lemma 1 shows that the necessary concentration is ensured by our invariance penalty. Experimentally we see this property holds in practise (figure 2).

C Additional Results

Proof of Lemma 1. Assume the data x is σ_x^2-sub-Gaussian. In practise this holds since x is bounded. It immediately follows that L-Lipschitz function $f(x)$ sub-Gaussian with parameter at most L. Now we will characterize the reduction in variance from x to f. Assume there is a ball of radius B around each point such that for any augmentation x_i^s of x_i $\|x_i - x_i^s\|_2^2 \leq B$. By assumption (A1) we have that $\|f(x_i) - f(x_i^s)\|_2^2 \leq \rho$. This implies that for points x_i and x_j such that $\|x_i - x_j\|_2^2 \leq 2B$, there exists a region of overlap so that $\|f(x_i) - f(x_j)\|_2^2 \leq \|f(x_i) - f(x_i^s)\|_2^2 + \|f(x_i^s) - f(x_j)\|_2^2 \leq 2\rho$.

In practise this says that there are augmentations of x_i which are sufficiently similar to augmentations of x_j so that their representations should be similar, thereby driving $f(x_i)$ and $f(x_j)$ to be closer.

The variance of points in f space is

$$\sigma_f^2 = \frac{1}{2N^2}\sum_i\sum_j \|f(x_i) - f(x_j)\|_2^2$$

The overlap $B < \|x_i - x_j\|_2^2 \leq 2B$ induces a graph where we say $j \in \mathcal{N}(i)\ \forall\ j$ s.t. $\|x_i - x_j\|_2^2 \leq 2B$. For N samples we can decompose the variance as

$$\begin{aligned}\sigma_f^2 &= \frac{1}{2N^2}\sum_i\sum_j \|f(x_i) - f(x_j)\|_2^2 \\ &= \frac{1}{2N^2}\sum_i\sum_{j\in\mathcal{N}(i)} \|f(x_i) - f(x_j)\|_2^2 + \sum_{j'\notin\mathcal{N}(i)} \|f(x_i) - f(x_{j'})\|_2^2\end{aligned}$$

By smoothness of f we always have that have $\|f(x_i) - f(x_{j'})\|_2^2 \leq L\|x_i - x_{j'}\|_2^2$. By the constraint we have that $\|f(x_i) - f(x_j)\|_2^2 \leq \frac{2\rho L}{B}\|x_i - x_j\|_2^2\ \forall j \in \mathcal{N}(i)$ and for $\delta = \frac{2\rho L}{B} < 1$.

Constant proportion overlap. Now, assuming that for each point i there is a constant proportion of the points, $0 \leq \alpha \leq 1$ in the set $\mathcal{N}(i)\ \forall i$ we can obtain the following inequality

$$\begin{aligned}\sigma_f^2 &= \frac{1}{2N^2}\sum_i\sum_j \|f(x_i) - f(x_j)\|_2^2 \\ &\leq \alpha\delta\sigma_x^2 + (1-\alpha)L\sigma_x^2 \\ &= (\alpha\delta + (1-\alpha)L)\sigma_x^2\end{aligned} \tag{7}$$

For $\sigma_f^2 \leq \sigma_x^2$ we require $(\alpha\delta + (1-\alpha)L) \leq 1$. Since both terms are positive we separately require $(1-\alpha)L \leq 1$:

$$\begin{aligned}(1-\alpha)L &< 1\\ (1-\alpha) &< \frac{1}{L}\\ \alpha &> (1-\frac{1}{L})\end{aligned}$$

This condition makes sense since the larger α, the fewer unconnected components in the graph. If the above holds, we also require $\alpha\frac{2\rho L}{B} < 1-(1-\alpha)L$ to ensure the sum is bounded above by 1. This implies $\rho < \frac{(1-(1-\alpha)L)B}{2L\alpha}$.

However, α is a property of the augmentation set and not directly a user-controllable parameter so if α is too small or the function is not smooth enough, it might not be possible to set ρ in such a way to induce contraction in σ_f^2.

In the next section we derive a tighter concentration based on the structure of random graphs which are induced by the connectivity between data points and their augmentations.

Random graphs. Consider the graph $G(V,E)$ induced by the constraints $(i,j) \in E \;\forall\; \|x_i - x_j\|_2^2 \leq 2B$. Call $\mathcal{N}(i)$ the set of neighbours of point i. For N points, if there is a constant probability α that $j \in \mathcal{N}(i)$ then $G_{N,\alpha}$ is an Erdös-Renyi graph.

From Theorem 3, if $\alpha \geq \frac{c\log N}{N}$ for $c > 1$ then with high probability, there are *no* unconnected components in G. That is, every vertex in V is reachable from any other vertex in a finite number of steps. We can then decompose the contribution to the variance in terms of components in the graph that are adjacent and those which are reachable within a certain number of steps.

Let the degree—the shortest path—between any two points be at most D we obtain the following refinement of eq. (7)

$$\begin{aligned}\sigma_f^2 &= \frac{1}{2N^2}\sum_i\sum_j \|f(x_i) - f(x_j)\|_2^2\\ &\leq \alpha\delta\sigma_x^2 + (1-\alpha)D\delta\sigma_x^2\end{aligned}$$

From Theorem 4 we have with high probability that $3 \leq D \leq 4$. So for $\sigma_f^2 \leq \frac{1}{\kappa}\sigma_x^2$ with $\kappa \geq 1$ we require $\rho \leq \frac{B}{2L\kappa(\alpha+3(1-\alpha))} \leq \frac{B}{6L\kappa}$. □

Theorem 3 (Connectedness (Erdős & Rényi, 1960))**.** *If $p = \frac{c\log n}{n}$ where $c > 1$ with high probability then the graph $G(n,p)$ has no unconnected components.*

Definition 1 (Diameter)**.** *For a connected graph, $G(V,E)$ the diameter diam$(G) = \max$ dist(v_i, v_j) where dist(v_i, v_j) is the minimum number of edges in the path between v_i and v_j.*

Theorem 4 (Diameter of random graphs (Frieze & Karoński, 2016))**.** *Let $d \geq 2$ be a fixed positive integer. For $c > 0$ and*

$$p^d n^{d-1} = \log(n^2/c)$$

Then diam$(G_{n,p}) \geq d$ with probability $\exp(-c/2)$ and diam$(G_{n,p}) \leq d+1$ with probability $1-\exp(-c/2)$.

D Generalizing Contrastive Learning

D.1 Refinements

On the unsupervised observed data $\mathcal{D}$, any task as defined by targets Y_t induces an equivalence relation, i.e. Y_t partitions $\mathcal{D}$ into equivalence classes. It divides $\mathcal{D}$ based on values of the target, $\mathcal{D} = \{\{x_a | y_a = y_i\}_{i=1}^M\}$ where $\{y_1, \ldots, y_M\}$ for some M is the set of target values. Here the equivalence relation associates datapoints based on the value of the target they predict. For example,

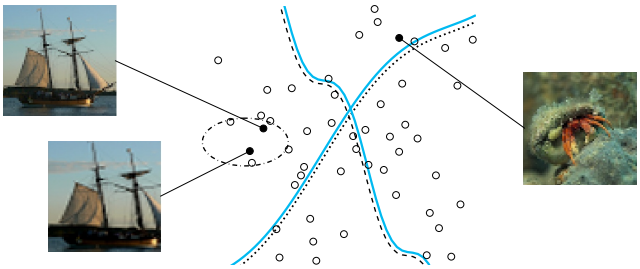

Figure 4: Visualization of a refinement of a set of tasks. The tasks are to classify aquatic vs non-aquatic life and animal vs non-animal with the individual class boundaries denoted by the dashed and dotted black lines. A refinement for these tasks is a to classify aquatic animal vs aquatic non-animal vs non-aquatic animal vs non-aquatic non-animal and the class boundaries are given in teal. The ellipse indicates the set of points induced by augmenting the image of the ship.

if $\mathcal{D}$ is a set of images of cats and dogs and Y_t denotes labels cat and dog, then $\mathcal{D}$ is partitioned into two equivalence classes corresponding to cat and dog images by Y_t.

Intuitively, a refinement is a subdivision of an existing partition. For a visualization of a refinement of a set of tasks see Figure 4. To mathematically define refinements, we first need to introduce what it means for an equivalence relation to be finer than another equivalence relation.

Definition 2. (Fineness). *Let $\sim$ and $\approx$ be two equivalence relations on the set $\mathcal{D}$. If every equivalence class of $\sim$ is a subset of an equivalence class of $\approx$, we say that $\sim$ is finer than $\approx$.*

Now we define what refinements.

Definition 3. (Refinement). *Let A, B be sets of equivalence classes induced by equivalence relations $\sim$ and $\approx$ over the set $\mathcal{D}$. If $\sim$ is finer than $\approx$, then we call A a refinement of B.*

Furthermore, we can relate the corresponding sets of equivalence classes.

Lemma 2. *Let $\sim$ and $\approx$ be two equivalence relationships on the set $\mathcal{D}$ and denote the corresponding induced partitions by A and B. If $\sim$ is finer than $\approx$, then every equivalence class of $\approx$ is a union of equivalence classes of $\sim$.*

Coming back to the example of cats and dogs, let $\approx$ be the relation that associates cats with cats and dogs with dogs. Now the relation $\sim$ which associated both cats and dogs with their specific breed (e.g. poodles with other poodles) is finer than $\approx$. Note that $\sim$ partitions $\mathcal{D}$ into breeds and so we can easily generate the sets of cats and dogs (i.e. equivalence classes of $\approx$) by taking a union over all the corresponding breeds.

D.2 Proof of Theorem 1

Definition 4. (Invariant Representation). *Let X and Y be the covariates and target, respectively. We call $f(X)$ an invariant representation for Y under style S if*

$$p^{do(S=s_i)}(Y \mid f(X)) = p^{do(S=s_j)}(Y \mid f(X)) \quad \forall\, s_i, s_j \in \mathcal{S}, \tag{8}$$

where $do(S = s)$ denotes assigning S the value s and $\mathcal{S}$ is the domain of S.

Theorem 1. Let $\mathcal{Y} = \{Y_t\}_{t=1}^{T}$ be a family of downstream tasks. Let Y^R be a refinement for all tasks in $\mathcal{Y}$. If $f(X)$ is an invariant representation for Y^R under changes in style S, then $f(X)$ is an invariant representation for all tasks in $\mathcal{Y}$ under changes in style S, i.e.

$$p^{do(s_i)}(Y^R \mid f(X)) = p^{do(s_j)}(Y^R \mid f(X)) \quad \Rightarrow \quad p^{do(s_i)}(Y_t \mid f(X)) = p^{do(s_j)}(Y_t \mid f(X)) \tag{9}$$

for all $t \in \{1, \ldots, T\}$ and for all $s_i, s_j \in \mathcal{S}$ with $p^{do(s_i)} = p^{do(S=s_i)}$. Thus, $f(X)$ is a representation that generalizes to $\mathcal{Y}$.

Proof. Let $t \in \{1, \ldots, T\}$. We have

$$p^{\mathrm{do}(s_i)}(Y_t|f(X)) = \int p^{\mathrm{do}(s_i)}(Y_t|Y^R)p^{\mathrm{do}(s_i)}(Y^R|f(X))dY^R = \int p(Y_t|Y^R)p^{\mathrm{do}(s_i)}(Y^R|f(X))dY^R$$
$$= \int p(Y_t|Y^R)p^{do(s_j)}(Y^R|f(X))dY^R = p^{do(s_j)}(Y_t|f(X)).$$

For the first equality, we used the fact that Y^R is a refinement of Y_t. To see this note that discrete random variables (e.g. Y^R and Y_t) induce equivalence relationships on the sample space, i.e. their events partition the sample space into equivalence classes. Since Y^R is a refinement of Y_t, we know that the equivalence relation induced by Y^R is finer than the equivalence relation induced by Y_t. By Lemma 2, we then know that every equivalence class induced under Y_t can be constructed from a union of equivalence classes induced under Y^R . Thus, we have that Y_t is a function of Y^R, i.e. $Y_t = g(Y^R)$ for some function g and thus we have that $p(Y_t|Y^R) = p(Y_t|Y^R, f(X))$. For the second and last equality, we used that the mechanism of $Y_t|Y^R$ is independent of S, i.e. $p^{\mathrm{do}(s_i)}(Y_t|Y^R) = p^{do(s_j)}(Y_t|Y^R)$. The third equality follows from the assumption that $f(X)$ is an invariant representation for Y^R under changes in S. Thus, we get that $f(X)$ is an invariant representation for Y_t under changes in S. Specifically, for a representation to be an invariant representation for Y_t it is a *sufficient condition* for it to be an invariant representation for Y^R. □

E Experimental Details

E.1 Image Augmentations

For pretraining the representations in ReLIC, we apply the augmentation scheme proposed in SimCLR (Chen et al., 2020a) and used in (Grill et al., 2020). This consists of the following augmentations applied in the order they are listed

- random crop – we randomly crop the image using an area randomly selected between 8% and 100% of the image with an logarithmically sampled aspect ration between $3/4$ and $4/3$. After this, we resize the patch to 224×224;
- random horizontal flip;
- color jittering – we apply in random order perturbations to brightness, contrast, saturation and hue of the image by shifting them by a random uniform offset;
- grayscale – we randomly apply grayscaling;
- Gaussian blurring – we blur the image using a 23×23 square Gaussian kernel with standard deviation uniformly sampled in $[0.1, 0.2]$;
- solarization – we transform all the pixels with $x \to x * 1_{\{x<0.5\}} + (1-x) * 1_{\{x \geq 0.5\}}$.

We use the same parameters for the augmentations and probabilities of applying individual augmentations as SimCLR (Chen et al., 2020a). After applying augmentations, we normalize the images with the mean and standard deviation computed on ImageNet across the color channels.

E.2 Architecture

We test ReLIC on two different architectures – ResNet-50 (He et al., 2016) and ResNet-50 with target network as in (Grill et al., 2020). For ResNet-50, we use version 1 with post-activation. We take the representation to be the output of the final average pooling layer, which is of dimension 2048. As in SimCLR (Chen et al., 2020a), we use a critic network to project the representation to a lower dimensional space with a multi-layer perceptron (MLP). When using ResNet-50 as encoder, we treat the parameters of the MLP (e.g. depth and width) as hyperparameters and sweep over them. This MLP has batch normalization (Ioffe & Szegedy, 2015) after every layer, rectified linear activations (ReLU) (Nair & Hinton, 2010). We used a 4 layer MLP with widths $[4096, 2048, 1024, 512]$ and output size 128 with ResNet-50. When using a ResNet-50 with target networks as in (Grill et al., 2020), we exactly follow their architecture settings.

E.3 OPTIMIZATION

We use a batch size of 4096 and the LARS optimizer (You et al., 2017) with a cosine decay learning rate schedule (Loshchilov & Hutter, 2017) for 1000 epochs with 10 epochs for warm-up. We exclude the biases and batch normalization parameters from LARS adaptation. We use as the base learning rate 0.3 for ResNet-50 and 0.2 for ResNet-50 with target network. We scale this learning rate by batch size$/256$ and use a global weight decay parameter of $1.5 * 10^{-6}$ and exclude the biases and batch normalization parameters. For the target network, we follow the approach of BYOL (Grill et al., 2020) and start the exponential moving average parameter τ at $\tau_{base} = 0.996$ and increase it to one during training via $\tau = 1 - (1 - \tau_{base})(\cos(\pi k/K) + 1)/2$ with k the current training step and K the maximum number of training steps.

E.4 EVALUATION ON IMAGENET

We follow the standard linear evaluation protocol on ImageNet as in (Kolesnikov et al., 2019; Chen et al., 2020a; Grill et al., 2020). We train a linear classifier on top of the fixed representation, i.e. we do not update the network parameters or the batch statistics. For training, we randomly crop and resize images to 224×224, and randomly horizontally flip the images after that. For testing, the images are resized to 256 pixels along the shorter dimension with bicubic resampling after which we take a center crop of size 224×224. Both for training and testing, the images are normalized by substracting the mean and standard deviations across the color channels computed on ImageNet after the augmentations. We use Stochastic Gradient Descent with a Nestorov momentum of 0.9 and train for 80 epochs with a batch size of 1024. We do not use any regularization techniques, e.g. weight decay.

E.5 ROBUSTNESS AND GENERALIZATION

E.5.1 DATASET DETAILS

ImageNet-C. The ImageNet-C dataset (Hendrycks & Dietterich, 2019) consists of 15 different types of corruptions from the noise, blur, weather, and digital categories applied to the validation images of ImageNet. This dataset is used for measuring semantic robustness. Figure 5 visualizes the corruption types. Each type of corruption has 5 levels of severity, i.e. there are 75 distinct corruptions in the dataset. In Figure 6, we display the Impulse noise corruption for 5 different severity levels. As can be seen, with increasing severity level the image becomes increasingly corrupted and difficult to parse. In addition to these 75 corruption types, there are an additional 4 corruption types (speckle noise, gaussian blur, spatter and saturate) that are provided as a validation set. We use these additional corruption types for selecting the best hyperparameters. For further details on this dataset, please refer to (Hendrycks & Dietterich, 2019).

ImageNet-R. The ImageNet-R dataset (Hendrycks et al., 2020) consists of $30,000$ images depicting various artistic renditions (e.g., paintings, sculpture, origami, cartoon) of 200 ImageNet object classes. This dataset is used to measure out-of-distribution generalization to various abstract visual renditions as it emphasizes shape over texture. The data was collected primarily from Flickr and also includes line drawings from (Wang et al., 2019). The images represent naturally occurring objects and have different textures and local image statistic to those of ImageNet. Figure 7 visualizes different images from the dataset. For further details on this dataset, please refer to (Hendrycks et al., 2020).

E.5.2 EVALUATION

To evaluate robustness and generalization of the learned representation, we follow the standard linear evaluation protocol on ImageNet as in (Chen et al., 2020b;a; Kolesnikov et al., 2019). We train a linear classifier on top of the frozen representation, i.e. we do not update either the network parameters nor the batch statistics. During training, we augment the data by randomly cropping, resizing to 224×224 and randomly flipping the image. At test time, images are resized to 256 pixels along the shorter side via bicubic resampling and we take a 224×224 center crop. Both during training and testing, after applying augmentations we normalize the color channels by subtracting the average color and dividing by the standard deviation that is computed on ImageNet. We optimize the cross-entropy loss using Stochastic Gradient Descent with Nestorov momentum of 0.9. We sweep

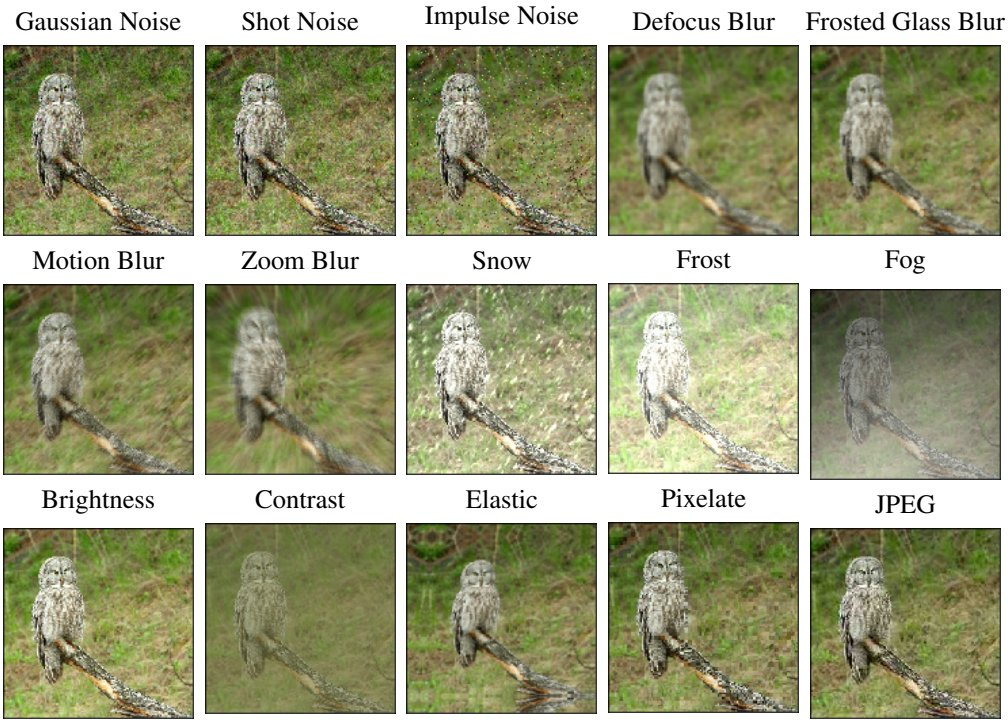

Figure 5: The ImageNet-C dataset consists of 15 types of corruptions from noise, blur, weather, and digital categories. Each type of corruption has five levels of severity, resulting in 75 distinct corruptions. See different severity levels in Figure 6.

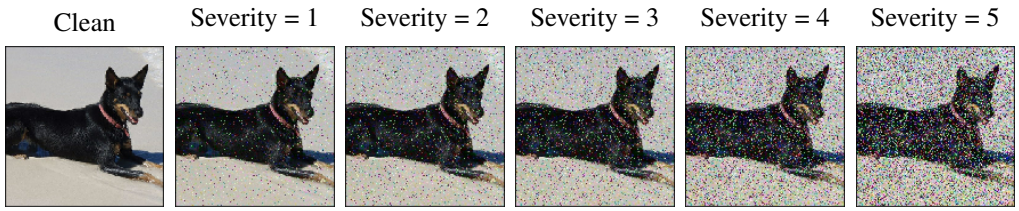

Figure 6: The 5 different levels of severity of Impulse noise corruption available in the ImageNet-C dataset. With increasing severity the dog image is markedly corrupted.

over number for epochs $\{30, 50, 60, 90\}$, learning rates $\{0.4, 0.3, 0.2, 0.1, 0.05, 0.01\}$ and batch sizes $\{1024, 2048, 4096\}$. We select hyperparameters on the validation set provided in ImageNet-C and report the performance on ImageNet-R and on the test set of ImageNet-C under the best validation hyperparameters. We do not use any regularization techniques such as weight decay, gradient clipping, $tanh$ clipping or logits regularization.

Figure 7: Example images from the dataset ImageNet-R which contains $30,000$ images of 200 ImageNet classes. This dataset emphasizes shape over texture and has different textures and local image statistic to those of ImageNet.

E.5.3 ROBUSTNESS METRICS AND FURTHER RESULTS

Let f be a classifier that has not been trained on ImageNet-C. For each corruption type c and level of severity $1 \leq s \leq 5$, denote the top-1 error of this classifier as $E^f_{s,c}$. Different corruption types pose different levels of difficulty. To make error rates across corruption types more comparable, the error rates are divided by AlexNet's errors. This standardized measure is the Corruption Error and is computed as

$$CE^f_c = \left(\sum_{s=1}^{5} E^f_{s,c}\right) / \left(\sum_{s=1}^{5} E^{AlexNet}_{s,c}\right)$$

The average error across all 15 corruption types is called the mean Corruption Error (mCE). Corruption Errors and mCE measure absolute robustness.

To better assess robustness, we also report the relative Corruption Error which measures relative robustness, i.e. loss in performance under corruptions. Denote by E^f_{clean} the top-1 error rate for f on the clean test set of ImageNet. The relative Corruption Error is given as

$$rCE^f_c = \sum_{s=1}^{5} \left(E^f_{s,c} - E^f_{clean}\right) / \sum_{s=1}^{5} \left(E^{AlexNet}_{s,c} - E^{AlexNet}_{clean}\right)$$

The mean relative Corruption Error (mrCE) is the mean of the relative Corruption Errors across all the corruption types. For more details and intuitions about there measures please refer to (Hendrycks & Dietterich, 2019).

In Table 6, we report Corruption Errors for Blur, Weather, and Digital corruption types. In Table 7, we report the relative robustness. As per (Hendrycks & Dietterich, 2019), we used the following values as the average AlexNet errors across severities, i.e. $\frac{1}{5}\sum_{s=1}^{5} E^{AlexNet}_{s,c}$, to normalize the Corruption Error values – Gaussian Noise 88.6%, Shot Noise 89.4%, Impulse Noise 92.3%, Defocus Blur 82.0%, Glass Blur 82.6%, Motion Blur 78.6%, Zoom Blur 79.8%, Snow 86.7%, Frost 82.7%, Fog 81.9%, Brightness 56.5%, Contrast 85.3%, Elastic Transformation 64.6%, Pixelate 71.8%, JPEG 60.7%, Speckle Noise 84.5%, Gaussian Blur 78.7%, Spatter 71.8%, Saturate 65.8%.

Table 6: Mean Corruption Error (mCE) and Corruption Error values for Blur, Weather, and Digital corruption types on ImageNet-C. All models are trained only using clean ImageNet images.

		Blur				**Weather**				**Digital**			
Method	mCE	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Supervised	76.7	75	**89**	**78**	**80**	78	75	66	57	71	**85**	77	77
Using ResNet-50:													
SimCLR	87.5	94.8	103.3	101.8	101.9	83.7	80.6	65.6	71.5	54	106.8	105.2	93
ReLIC	76.4	81.4	96.9	92.7	93.2	73.7	71.2	54.5	60.2	**46.9**	97.4	85.5	77.2
ResNet-50 with target network:													
BYOL	72.3	75	93.6	86.3	87.9	74.3	69.1	48.5	55	48.6	90.4	**74.3**	73
ReLIC	**70.8**	**73.2**	94	81.9	87	**73.2**	**68**	**47.5**	**54.2**	48.4	89.5	75.6	**71.8**

Table 7: Mean relative Corruption Error (mrCE) and relative Corruption Error values for different corruptions and methods on ImageNet-C. The mrCE value is the mean relative Corruption Error of the corruptions in Noise, Blur, Weather, and Digital columns. All models are trained only using clean ImageNet images. RELIC-t denotes using RELIC with a ResNet-50+target network architecture as in BYOL (Grill et al., 2020).

		Noise			**Blur**				**Weather**				**Digital**			
Method	mrCE	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
Supervised	105	104	107	107	97	**126**	**107**	**110**	101	97	79	62	89	**146**	111	132
Using ResNet-50:																
SimCLR	111.9	88	92.7	106.6	122.2	139.6	140.5	139.4	96.9	91.6	59.9	74.8	36.7	181.5	158.3	149.8
ReLIC	**87.7**	**67.3**	73.1	84.8	96.1	128.7	123	123.1	**79.3**	**74.4**	38.9	**33.4**	**24.6**	157.5	112	**99.8**
ResNet-50 with target network:																
BYOL	90	72.5	77.2	86.7	93	132	120	122.5	89.7	80.2	36.6	41.5	37.8	155	**97.6**	108.4
ReLIC	88.4	69.1	**73**	**79.2**	**90.4**	134.2	111.6	121.9	88.5	79.2	**35.6**	41.7	38.5	154.5	102.7	106.8

E.6 EVALUATION ON ATARI

For our experiments on Atari, we use the agent from R2D2 (Kapturowski et al., 2019) with standard hyperparameters noted below. We train each agent on approximately 15 billion frames and add a

second encoder with the same architecture used in the Q-Network of the original agent. This second encoder is trained with a separate optimizer with only a representation learning objective. The agent then takes the output of this encoder as a given input. We use standard augmentations used in prior work (Kostrikov et al., 2020) where we pad the frames on all sides with 4 pixels copied from the borders and then randomly cropping 84 windows. We randomly shift pixel intensity according to the distribution $s = 1.0 + 0.1 * \mathcal{N}'$ where $\mathcal{N}'$ is the standard Normal distribution with values clipped between -2 and 2. s is then multiplied by the original image to return the augmented image.

RELIC and SimCLR For our implementation of RELIC and SimCLR, we do not use a critic embedding at all and utilize the last layer of the encoder for the objective. As in CURL (Srinivas et al., 2020) we utilize a target encoder for the second augmentation where we update the weights with a momentum of .99. We also clipped the gradients of our optimizer using a global norm ratio of 40. We report the hyperparameters in Table 8

Table 8: RELIC and SimCLR Details

Parameter	Value
Normalize Inputs	True
Temperature	1.0 Constant
Scaling of Embeddings	False
Optimizer	Adam
Learning Rate	5e-4
Epsilon	0.01
Beta 1	0.9
Beta 2	0.999

CURL For CURL, we use a second encoder as noted before. With the exception of the encoder architecture and the optimizer parameters, all hyperparameters are the same as in (Srinivas et al., 2020) including the momentum value for the target network weight updates. We utilize the same architecture in the paper with a linear layer as a critic embedding for the target encoder.

Table 9: CURL Details

Parameter	Value
Optimizer	Adam
Learning Rate	1e-3
Epsilon	0.01
Beta 1	0.9
Beta 2	0.999

BYOL In BYOL, we utilize two-layer perceptron networks as our predictor and projection layers. For both networks, the number of hidden units in the two layers was 1024 and 512. We use a target network update momentum of .99. The optimizer parameters are the same as in Table 8.

Direct Augmentation We also compared against direct augmentation of the observations in the replay buffer as in DrQ (Kostrikov et al., 2020). We keep the architecture the same in this instance and use two duplicate encoders as input to the agent. In this case, the optimizer can jointly update both encoders and train them end-to-end.

Table 10: Individual Mean Episode Return on Atari.

Games	Average Human	Random	RELIC	SimCLR	CURL	BYOL	Augmentation
alien	7127.70	227.80	8766.57	**10082.54**	8506.48	9671.89	5201.93
amidar	1719.50	5.80	**28449.26**	28141.18	27213.75	25965.05	867.66
assault	742.00	222.40	**92963.07**	36109.84	7139.67	13565.20	1539.71
asterix	8503.30	210.00	**998426.72**	997305.51	661431.39	986307.92	26239.64
asteroids	47388.70	719.10	83669.38	7299.90	76612.17	55936.02	**101340.17**
atlantis	29028.10	12850.00	1575940.94	1584392.76	**1584698.01**	1530122.45	794011.79
bank heist	753.10	14.20	1521.38	2467.62	**4095.29**	1659.94	771.60
battle zone	37187.50	2360.00	**452831.48**	278903.14	287792.06	338695.47	31511.75
beam rider	16926.50	363.90	**136695.24**	98551.42	116794.58	87454.20	46894.14
berzerk	2630.40	123.70	**146213.60**	1301.36	73754.38	1265.21	73645.52
bowling	160.70	23.10	205.09	193.50	**230.31**	172.21	164.68
boxing	12.10	0.10	100.00	100.00	100.00	100.00	**100.00**
breakout	30.50	1.70	405.05	404.06	407.14	**409.48**	150.67
centipede	12017.00	2090.90	**220886.86**	99544.92	167779.11	146735.67	20152.01
chopper command	7387.80	811.00	999900.00	999900.00	**999900.00**	962003.61	5399.56
crazy climber	35829.40	10780.50	272179.68	266870.81	**301689.62**	210477.39	96538.00
defender	18688.90	2874.50	**576405.57**	522617.05	560816.84	493410.36	78750.19
demon attack	1971.00	152.10	143774.79	**143786.19**	143737.36	143574.86	821.98
double dunk	-16.40	-18.60	**24.00**	24.00	24.00	24.00	14.82
enduro	860.50	0.00	2371.27	2366.19	**2373.12**	2368.00	1361.66
fishing derby	-38.70	-91.70	68.17	**83.00**	72.21	70.11	19.93
freeway	29.60	0.00	33.00	32.93	**33.04**	33.00	32.00
frostbite	4334.70	65.20	10156.41	**11171.49**	3693.20	5793.80	5708.35
gopher	2412.50	257.60	**123170.74**	122368.21	122371.64	120317.04	43711.82
gravitar	3351.40	173.00	4186.09	3601.14	**4997.87**	4048.25	2014.59
hero	**30826.40**	1027.00	13615.35	13523.98	13620.78	13558.04	8957.00
ice hockey	0.90	-11.20	56.39	48.27	45.06	**59.70**	-2.43
jamesbond	302.80	29.00	**15632.87**	5714.62	10052.04	10099.81	1441.95
kangaroo	3035.00	52.00	14342.59	14215.11	11674.19	**14471.65**	7249.73
krull	2665.50	1598.00	**137099.65**	100426.69	86049.99	80414.04	16626.09
kung fu master	22736.30	258.50	**230241.57**	220076.57	228943.94	208064.38	64632.42
montezuma revenge	**4753.30**	0.00	1066.67	733.33	1072.30	419.54	26.67
ms pacman	6951.60	307.30	13367.55	12053.76	**13465.80**	12726.79	3238.90
name this game	8049.00	2292.30	**48669.30**	46657.55	47417.82	44848.29	13416.57
phoenix	7242.60	761.40	**803108.37**	253542.40	580969.56	20317.80	6264.39
pitfall	**6463.70**	-229.40	0.00	0.00	0.00	0.00	0.00
pong	14.60	-20.70	**21.00**	21.00	21.00	21.00	21.00
private eye	**69571.30**	24.90	10154.93	5115.34	5190.28	470.68	111.77
qbert	13455.00	163.90	**353197.13**	24340.75	208207.97	57261.24	11051.97
riverraid	17118.00	1338.50	**23525.44**	20400.83	20230.02	22206.57	10487.59
road runner	7845.00	11.50	213173.15	236235.30	241917.98	238880.54	**440430.17**
robotank	11.90	2.20	97.65	82.60	**98.13**	62.54	49.98
seaquest	42054.70	68.40	**999999.00**	999999.00	666700.67	29160.93	37397.26
skiing	**-4336.90**	-17098.10	-24761.06	-23076.73	-15497.66	-26028.08	-22162.91
solaris	**12326.70**	1236.30	4594.37	4571.27	4276.39	4331.03	4142.69
space invaders	1668.70	148.00	**3625.52**	3619.94	3542.48	3613.93	835.37
star gunner	10250.00	664.00	283499.72	**289099.89**	129720.84	175486.67	43167.07
surround	6.50	-10.00	**10.00**	9.96	1.60	9.56	-0.64
tennis	-8.30	-23.80	0.00	0.00	0.00	0.00	**0.12**
time pilot	5229.20	3568.00	309297.74	92888.66	**400326.69**	48011.44	14198.37
tutankham	167.60	11.40	**371.17**	306.45	337.61	285.36	144.30
up n down	11693.20	533.40	**577256.03**	520666.59	566912.89	552110.67	143512.38
venture	1187.50	0.00	1929.53	**1945.20**	1906.84	1881.76	733.29
video pinball	17667.90	0.00	978292.52	**993332.08**	932523.58	623223.24	37584.71
wizard of wor	4756.50	563.50	**123513.74**	89462.62	106801.20	68256.44	5940.82
yars revenge	54576.90	3092.90	228704.52	99636.25	**229221.52**	86847.75	48041.63
zaxxon	9173.30	32.50	**120830.77**	57379.66	85906.74	48067.61	23688.22