Approximate Bijective Correspondence for isolating factors of variation

Anonymous authors
Paper under double-blind review

Abstract

Representational learning forms the backbone of most deep learning applications, and the value of a learned representation is intimately tied to its information content regarding different factors of variation. Finding good representations depends on the nature of supervision and the learning algorithm. We propose a novel algorithm that relies on a weak form of supervision where the data is partitioned into sets according to certain *inactive* factors of variation. Our key insight is that by seeking approximate correspondence between elements of different sets, we learn strong representations that exclude the inactive factors of variation and isolate the *active* factors which vary within all sets. Importantly, the information isolated is complementary to that of most other contrastive learning approaches, which isolate the inactive factors of variation. We demonstrate that the method can work in a semi-supervised scenario, and that a portion of the unsupervised data can belong to a different domain entirely. Further control over the content of the learned representations is possible by folding in data augmentation to suppress nuisance factors. We outperform competing baselines on the challenging problem of synthetic-to-real object pose transfer.

1 Introduction

A good representation is just as much about what it excludes as what it includes, in terms of the factors of variation across a dataset (Tian et al., 2020b). Control over the information content of learned representations depends on the nature of available supervision and the algorithm used to leverage it. For example, complete supervision of the desired factors of variation provides maximum flexibility for fully disentangled representations, as it is straightforward to obtain an interpretable mapping between elements and the factors of variation (Bengio et al., 2013; Higgins et al., 2018). However, such supervision is unrealistic for most tasks since many common factors of variation in image data, such as 3D pose or lighting, are difficult to annotate at scale in real-world settings. On the other hand, unsupervised learning makes the fewest limiting assumptions about the data but does not allow control over the discovered factors. Neither extreme, fully supervised or unsupervised, is practical for many real-world tasks.

As an alternative, we consider only weak supervision in the form of set membership (Kulkarni et al., 2015; Denton & Birodkar, 2017). Specifically, set supervision assumes that we can access subsets of training data within which some *inactive* factors of variation have fixed values and the remaining *active* factors freely vary. In many complex regression tasks that are beyond the scope of categorical classification, set supervision serves as a more flexible framework for operating on factors of variation across a dataset.

In the context of set supervision, existing methods can learn representations which isolate the inactive factors (Chen et al., 2020b; von Kügelgen et al., 2021). Consider the task of isolating 3D pose from images of cars. If the images could be grouped by pose (i.e. the inactive factor in each set is pose), then the training objective is straightforward – since each set contains images with identical poses, these images should be nearby in the representation space. However, in this scenario and more generally, this variant of set supervision is often prohibitive to obtain – in our example it requires identifying images of different cars from exactly the same viewpoint.

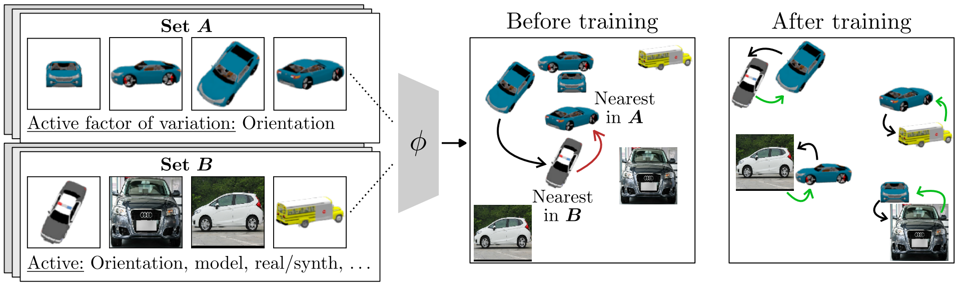


Figure 1: ***Approximate bijective correspondence (ABC).*** *Leveraging weak set supervision, ABC isolates factors of variation which actively vary across sets. Establishing one-to-one correspondence between sets of inputs requires isolating factors that commonly vary within each set and suppressing the factors which do not. For example, the images in set $\mathcal{A}$ (left) actively vary by only the orientation of the rendered car. We claim that if one-to-one correspondence can be found between $\mathcal{A}$ and $\mathcal{B}$, for all such set $\mathcal{A}$ and $\mathcal{B}$ pairs, it must leverage orientation. We find this to be true even when only one of the sets in each mini-batch is set-supervised, as above. Importantly, this allows the incorporation of out-of-domain data with no supervision at all, such as the images of real cars in $\mathcal{B}$. By training a neural network with a soft correspondence loss in representation space (middle), the learned representations (right) isolate the active factor of variation.*

A more readily available form of set supervision is where the desired factor is active in each set. Continuing the example, such supervision can easily be obtained by simply imaging each car from multiple viewpoints (set $\mathcal{A}$ in Figure 1). This does not require correspondence in viewpoints across object instances, nor any pose values attached to the images. However, this supervision makes learning much harder, as it no longer provides images which correspond in the desired factor.

In this work, our challenge is to isolate the **active** factors of variation given set supervision. We propose a novel approach, *approximate bijective correspondence* (ABC), based on finding correspondence between sets. The process of finding correspondences encourages isolating the active factors: to consistently match across sets, learned representations must ignore invariant information within a set (inactive factors) and focus on active factors common to both sets. Because the goal of learning is the active factors of variation common to all sets, a powerful consequence is that we are able to incorporate sets with extra active factors, including wholly unsupervised and even out-of-domain data (e.g., set $\mathcal{B}$ in Figure 1). For example, ABC-learned embeddings trained on sets of images as in Figure 1 isolate orientation, the common active factor across every pair of sets during training.

In our approach, corresponding points between sets are formed with a differentiable form of nearest neighbors (Goldberger et al., 2004; Movshovitz-Attias et al., 2017; Rocco et al., 2018; Snell et al., 2017; Dwibedi et al., 2019), and serve as positive pairs for use in a standard InfoNCE loss (van den Oord et al., 2019). We posit that the same desirable properties of learned representations that optimize InfoNCE on explicitly provided positive pairs – namely, *alignment*, where differences within positive pairs are ignored, and *uniformity*, where maximal remaining information is retained (Wang & Isola, 2020; von Kügelgen et al., 2021) – can be utilized to guide a network to find useful correspondences on its own. We find this to be true, with the important consequence that the information isolated is complementary to what would result from using the set-supervision directly as positive pairs.

The highlights of this work are the following: We demonstrate the strengths and limitations of ABC, and specify the defining properties of the learned representations. We quantitatively define the factor isolation in ABC-learned representations through mutual information measurements, leveraging complete knowledge of the generative factors in the synthetic Shapes3D dataset (Burgess & Kim, 2018) (Section 4.1). We connect properties of set supervision with their effects on the representations – namely that there is more freedom in the formation of sets than previously understood, and that larger sets yield more informative representations. ABC requires orders of magnitude fewer training steps than related methods to isolate handwriting style from digit identity (Section 4.2). Finally, ABC accomplishes the challenging real-world task of 3D object pose estimation (Section 4.3), through a training process that combines set-supervised synthetic data and unsupervised real images to generalize pose information across the category level and the synthetic-to-real domain gap.

2 Related work

Isolating factors of variation. Recent work (Locatello et al., 2019) has shown unsupervised disentanglement of latent factors to be impossible without incorporating some sort of supervision or inductive bias, spurring research into the best that can be achieved with different forms of supervision (Shu et al., 2020; von Kügelgen et al., 2021). A more realistic goal is the isolation of a subset of factors of variation, where learned representations are informative with respect to those factors of variation and not others, with no guarantees about the structure of these factors in latent space.

Set supervision. Often, data is readily grouped into sets according to certain factors of variation, without requiring explicit annotation on the factors. Generally, the methods harnessing information present in such groupings either **(i)** learn all factors and partition the representation such that one part is invariant across sets and the remaining part captures the intra-set (*active*) variation (Kulkarni et al., 2015; Mathieu et al., 2016; Cohen & Welling, 2015; Sanchez et al., 2020; Jha et al., 2018; Bouchacourt et al., 2018), or **(ii)** learn the factors which are invariant (*inactive*) across sets (Chen et al., 2020b; Tian et al., 2020b;a). The methods of **(i)** almost always employ generative models, with the exception of Sanchez et al. (2020), which grants it 6× faster training over the VAE-based approach of Jha et al. (2018); the downside is Sanchez et al. (2020) require seven networks and a two-stage, adversarial training process to learn first the inactive and then the active partitions of the representation. The methods of **(ii)** generally create subsets of data via augmentation (Chen et al., 2020b; He et al., 2020) or pretraining tasks (Misra & van der Maaten, 2019), or leverage multiple views of the same scene (Sermanet et al., 2018; Tian et al., 2020a), where semantic information is taken to be invariant across sets and is the target of training. By contrast, ABC directly learns *active* factors of variation across sets, offering a faster and simpler alternative to methods in **(i)** and tackling problems which are currently unassailable by methods in **(ii)**.

Videos, images, and point clouds are common forms of data which easily offer set supervision. Approaches to find correspondence between frames of related videos, first using a discrete form of cycle consistency (Aytar et al., 2018) and later a differentiable form Dwibedi et al. (2019), helped inspire this work. The latter relied on a soft nearest neighbor mapping, as has been used previously (Goldberger et al., 2004; Movshovitz-Attias et al., 2017; Rocco et al., 2018; Snell et al., 2017) and which our method uses as the first step to correspondence. Cycle consistency has also been used to establish point correspondences in images (Zhou et al., 2016; Oron et al., 2016) and 3D point clouds (Yang et al., 2020; Navaneet et al., 2020; Neverova et al., 2021). In contrast to methods focusing on specific applications such as action progression in videos (Dwibedi et al., 2019; Haresh et al., 2021) or robotics simulations (Zhang et al., 2021), we present a general approach applicable to a larger class of problems.

Characterization of learned representations. Fundamental questions around properties of good representations (Bengio et al., 2013; Tian et al., 2020b) and of disentanglement (Higgins et al., 2018; Locatello et al., 2019) remain actively debated. We employ MINE (Belghazi et al., 2018) to estimate the information content of known generative factors in ABC-learned representations, and corroborate the results using a classification task in Appendix A.

Most commonly, the quality of representations learned without explicit supervision is characterized by performance on a downstream task, for which we use the challenging real-world problem of 3D object pose estimation. Several pose estimation methods use pose-aware representations to tackle challenges like object symmetries; evaluation at test time then employs a codebook of images with known pose (Sundermeyer et al., 2018; 2020; Corona et al., 2018; Okorn et al., 2020). We probe the effectiveness of ABC at isolating pose when it is the active factor of variation, by training without annotations and then using a codebook constructed from unseen, out-of-domain images to evaluate.

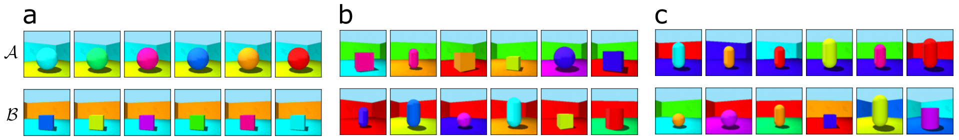
3 Algorithm

ABC uses set-supervised data, such that set membership is defined based on certain inactive factors; e.g., the data is grouped into sets such that all images in a set have the same object class, making the object class as the inactive factor. The basic idea of ABC is to consider minibatches with two sets of images, establish approximate bijective correspondences among their elements through the learned representations, and use the correspondences as positive pairs in the InfoNCE loss (van den Oord et al., 2019) to learn the visual representations.

To illustrate this better, let us consider the pose isolation task introduced earlier. Let us assume that a latent description consists of the make and model of the car, all specifics relating to color and structure, and the pose from which the image is captured. If we have set-supervised data where the car instance specifics are the inactive factors within each set and the only active factor is pose (e.g., Set $\mathcal{A}$ in Figure 1), ABC will pair elements across two sets such that pose is invariant for each pair. By training with these as positive pairs in the InfoNCE loss, the embeddings learned through ABC would isolate the active factor of pose.

Setup and notation: We follow the setup and notation from von Kügelgen et al. (2021), that uses a latent variable model for the theoretical modeling of self-supervised learning methods. Let us denote the input images as x from the observation space $\mathcal{X}$ and an associated latent code as z from the representational space $\mathcal{Z}$. As per the latent variable model, the observations can be generated from the latent code using an invertible function $x = f(z)$, with $z \sim p_z$. Without loss of generality, we assume that the the latent vector z can be partitioned into inactive z_i and active z_a components such that all elements within each set share identical z_i. Let $\phi(x) : \mathcal{X} \to \mathbb{R}^E$ be the function that maps the input vector to an embedding u in E-dimensional space. Our goal is to learn this function so that u may be informative with respect to one of the partitions of the true underlying latent code z.

Mini-batch construction: We either leverage natural groupings of images or curate images into sets by controlling for certain factors of variation during mini-batch construction, where each mini-batch consists of two such sets. For example, in Figure 2, we show example sets with different active and inactive factors of variation curated from the Shapes3D dataset (Burgess & Kim, 2018). Values for the inactive factors are randomly sampled and held fixed for each set, with the active factors free to vary (Figure 2a,b).



	Inactive	Active	Inactive	Active	Inactive	Active
$\mathcal{A}$	WH, FH, Sc, Sh, Or	OH	WH	OH, FH, Sc, Sh, Or	FH, Sh	WH, OH, Sc, Or
$\mathcal{B}$	WH, FH, Sc, Sh, Or	OH	WH	OH, FH, Sc, Sh, Or	$\varnothing$	WH, FH, OH, Sc, Sh, Or

WH = Wall hue   OH = Object hue   FH = Floor hue   Sc = Scale   Sh = Shape   Or = Orientation

Figure 2: ***Set supervision scenarios amenable to ABC.*** ***(a)*** *In the simple case with five inactive factors for each set, there is only one factor to isolate: the object hue.* ***(b)*** *The sets can be much less constrained, here defined by only a single inactive factor.* ***(c)*** *One set may be entirely unconstrained, with no inactive factors at all. In all three scenarios, ABC isolates factors which actively vary in both sets.*

Approach: Let the sets for a particular mini-batch be given by $\mathcal{A} = \{a_1, \ldots, a_n\}$ and $\mathcal{B} = \{b_1, \ldots, b_m\}$, respectively. Let us denote the associated embeddings as $\mathcal{U} = \{u_1, \ldots, u_n\}$ and $\mathcal{V} = \{v_1, \ldots, v_m\}$, where $u_i = \phi(a_i, w)$ and $v_i = \phi(b_i, w)$. Functionally, we parameterize ϕ with the same neural network (with weights w) for both A and B. Let $s(u, v)$ denote a similarity metric between points in embedding space, with $s(u, v) = s(v, u)$. To create an end-to-end differentiable loss, we use the soft nearest neighbor (Goldberger et al., 2004; Movshovitz-Attias et al., 2017; Rocco et al., 2018; Snell et al., 2017; Dwibedi et al., 2019).

Definition 1 (***Soft nearest neighbor***) *Given a point u and a set of points $\mathcal{V} = \{v_1, \ldots, v_m\}$, the soft nearest neighbor of u in the set V is given by $\tilde{u} = \sum_{j=1}^{m} \alpha_j v_j$, where $\alpha_j = \frac{\exp(s(u_i, v_j)/\tau)}{\sum_{k=1}^{m} \exp(s(u_i, v_k)/\tau)}$ and τ is a temperature parameter.*

We first compute the soft nearest neighbor for each $u_i \in \mathcal{U}$ as $\tilde{u}_i = \sum_{j=1}^{m} \alpha_j v_j$. A soft bijective correspondence between the two sets is quantified through an InfoNCE loss (van den Oord et al., 2019), averaged over every element in each of the sets.

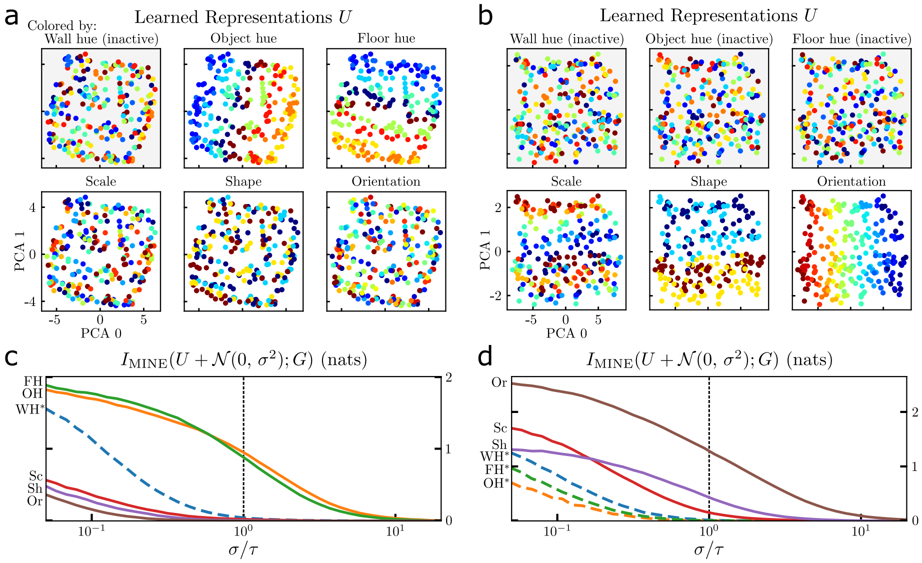


Figure 3: ***Isolation of active factors, Shapes3D.*** ***(a)*** *Trained with wall hue as the only inactive factor, information about object and floor hue is visually apparent in the first two principal components (>0.98 of total variance) of the $\mathbb{R}^{64}$ embeddings. Each scatter plot displays the same 256 embeddings, colored according to each generative factor.* ***(b)*** *With all hue factors inactive, the representations become informative about the geometric factors.* ***(c,d)*** *For the networks in* ***(a,b)****, respectively, we estimate the mutual information $I(U;G)$ between the representations and each of the generative factors using MINE (Belghazi et al., 2018). We add Gaussian noise to the representations to probe information content over different length scales in representation space. When σ equals the length scale of the loss (vertical dotted line), information about inactive factors (dashed) disappears.*

Definition 2 **(*Correspondence loss*)** *The correspondence loss from $\mathcal{U}$ to $\mathcal{V}$ is given by $\mathcal{L}(\mathcal{U},\mathcal{V}) = -\frac{1}{n}\sum_i^n \log\frac{\exp(s(u_i,\tilde{u}_i)/\tau)}{\sum_j^n \exp(s(u_j,\tilde{u}_i)/\tau)}$. The full loss is the sum, $\mathcal{L} = \mathcal{L}(\mathcal{U},\mathcal{V}) + \mathcal{L}(\mathcal{V},\mathcal{U})$.*

The temperature parameter τ sets a length scale in embedding space as the natural units for the loss. It is unimportant when using an unbounded similarity metric such as negative Euclidean distance, in which case a larger value of τ will lead to embeddings which are spaced further apart. By contrast, a metric like cosine similarity benefits from tuning τ.

ABC versus contrastive learning: While both ABC and self-supervised learning (SSL) methods such as SimCLR (Chen et al., 2020b) use the InfoNCE loss on positive and negative pairs, a fundamental difference arises from how one acquires the positive and negative pairs. In SSL the positive pairs are explicitly obtained through augmentations known to only affect certain 'style' variables, leaving 'content' invariant. In ABC, the positive pairs are unknown *a priori* and obtained through matching or finding nearby embeddings that possess similar values for some of the active factors. In many of our experiments, the inactive factors relate to content and active factors relate to style. However, ABC does not learn representations that isolate content or class information; rather, ABC isolates the active factors, i.e., style.

Double augmentation: We introduce a modification to the correspondence loss in order to suppress factors of variation which we can augment (e.g., translation and recoloring). With inspiration from (Chen et al., 2020b), we assume a group of transforms H is known from prior knowledge of the data, which leaves desired factors of variation unchanged (Higgins et al., 2018; Chen et al., 2020a). We randomly sample two transforms $h \in H$ per image per training step. Let $u_i^{(1)} = \phi(h^{(1)}a_i, w)$ and similarly for $u_i^{(2)}$. Then the correspondence loss becomes $\mathcal{L}(\mathcal{U},\mathcal{V}) = -\frac{1}{n}\sum_i^n \log\frac{\exp(s(u_i^{(1)},\tilde{u}_i^{(2)})/\tau)}{\sum_j^n \exp(s(u_j^{(1)},\tilde{u}_i^{(2)})/\tau)}$. In this manner we are able to exclude both the inactive factors across sets *and* augmentable factors of variation, granting more control over

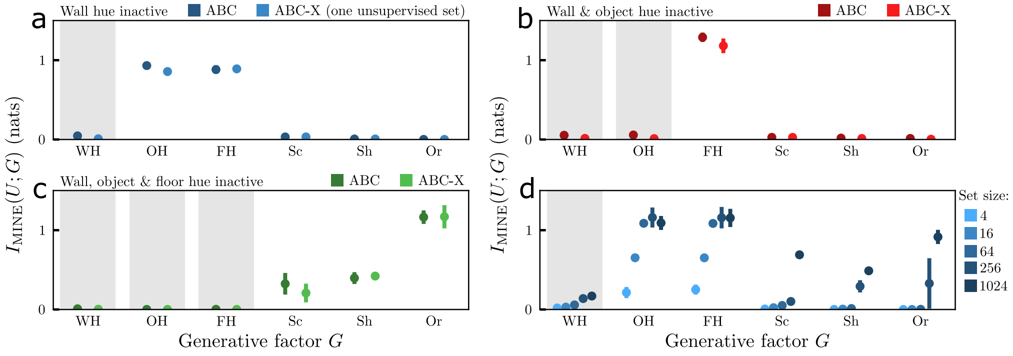


Figure 4: ***Influence of different aspects of set supervision.*** *We measure the information content of the learned representations U as in Figure 3, with added noise of magnitude $\sigma = \tau$. Error bars display the standard deviation across ten random seeds. The inactive factors during training are indicated by shading.* ***(a-c)*** *We find the isolation of active factors to be unchanged when training with one of the two sets unsupervised (ABC-X).* ***(d)*** *Increasing the set size isolates more of the active factors of variation because finding correspondence requires more discerning power.*

the nature of the learned representations. This can help prevent the correspondence task from being solved via 'shortcuts' through easy factors of variation (Tian et al., 2020b).

Incorporation of unsupervised data: Only the active factors of variation common to both sets are useful for establishing correspondence. One set's inactive factor of variation cannot help distinguish between elements of that set and therefore cannot help form correspondence with the elements of another, even if the factor actively varies in the second set. This has the powerful, previously unknown consequence (to our knowledge) that ABC can work just as well when one of the sets in each mini-batch is completely unconstrained, as in Figure 2c and Figure 1. *Wholly unsupervised, and even out-of-domain data with additional active factors due to domain differences, can be used in training.* We denote this version of the method ABC-Extraneous, or *ABC-X*, for utilizing data with extraneous active factors of variation.

Evaluation: We estimate the mutual information $I(U; G)$ between representations U and known latent factors G using mutual information neural estimation (MINE) (Belghazi et al., 2018). In general, deterministic networks fully preserve information between input and output, so noise is added for a meaningful quantity $I(U + \eta; G)$, with $\eta \sim \mathcal{N}(0, \sigma^2)$ (Saxe et al., 2019; Elad et al., 2019). This has the effect of excluding information at length scales smaller than σ from the information measurement. In the case where $s(u, v)$ is negative Euclidean distance, τ serves as a natural length scale of the correspondence loss so we use $\sigma = \tau$ when not sweeping over length scales, as in Fig. 3c,d (further discussion in App. A).

4 Experiments

We probe the method in three arenas. In the first, we leverage complete knowledge of generative factors in the artificial Shapes3D dataset (Burgess & Kim, 2018), in order to 1) experiment with different versions of set supervision, and 2) measure the information content of the learned representations and precisely illustrate the resultant factor isolation. Next, we demonstrate one significant practical advantage of ABC – speed – by isolating style from class of MNIST (LeCun & Cortes, 1998) digits. Finally, we apply ABC on a challenging pose estimation task, in both the real and synthetic domain, in the case where there are no pose annotations during training. Implementation details can be found in Appendix G.

4.1 Systematic Evaluations on Shapes3D

Images from the Shapes3D dataset consist of a geometric primitive with a floor and background wall (See Figure 2). There are six factors of variation in the dataset: three color factors (wall, object and floor hue) and three geometric factors (scale, shape and orientation).

Two examples of ABC-trained embeddings are shown in Figure 3, with the information content about active factors shown qualitatively (Fig. 3a,b) and quantitatively (Fig. 3c,d).

Average information measurements for many different training scenarios, over ten runs each, are shown in Figure 4. We discuss noteworthy aspects of ABC-trained representations below.

Information about active factors stored at larger length scales: In Figure 3c,d, information about the inactive factors of variation is suppressed and information about a subset of active factors is isolated over length scales relevant to the correspondence loss ($\sigma \sim \tau$). Factor isolation by ABC is therefore a separation of length scales, which makes the learned representations well-suited for tasks involving lookup (Section 4.3): the similarity of representations is governed most by the large length scales over which the active factors are informative.

Inactive factors always suppressed, subset of active factors isolated: In Figure 4 information with respect to the inactive factors in each set is always suppressed, though not all active factors are isolated. Only when all three hue factors are inactive (Fig. 4c) are the geometric factors present in the learned representations, seemingly because the 'easy' hue factors have all been suppressed. A similar differentiation between factors was noted in Tian et al. (2020b), where the authors suggested one factor of variation offered a "shortcut" for solving the contrastive learning task so the network could ignore a different factor.

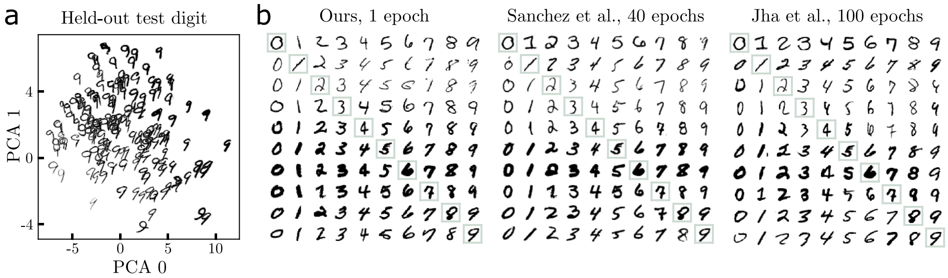


Figure 5: ***Fast style isolation on MNIST.*** *After training ABC with set supervision where digit class is the inactive factor, we evaluate the isolation of the factors of variation relating to style.* ***(a)*** *We display embeddings of the digit 9, held out during training to test the isolation of class-independent style information. The embeddings fan out by thickness and slant.* ***(b)*** *We perform retrieval on the test set using the boxed images along the diagonal as queries; the other images in each row are the nearest embeddings for each digit class. ABC retrieves images closer in style, more than an order of magnitude faster than the discriminative approach of Sanchez et al. (2020) and the VAE-approach of Jha et al. (2018).*

Semi-supervised is just as effective: Correspondence is found through active factors common to both sets, which means if one set consistently has additional active factors, they will not be useful for optimizing the ABC loss. In semi-supervised scenarios with one set-supervised set per mini-batch and the other consisting of random samples over the entire dataset (e.g., Fig. 2c), ABC-X is as performant as ABC with full set supervision (Fig. 4a-c).

Increasing set size isolates more active factors: Intuitively, finding a one-to-one correspondence between sets with more elements requires more discerning power. The measurements of $I(U;G)$ in Figure 4d show that as the set size climbs to 64, increasing information is gleaned about the two active hue factors. With a set size of 1024, all five active factors are isolated in the learned representations. Given that the set size effectively serves as the number of negative samples in the InfoNCE loss, and that more negative samples benefits contrastive learning (Hjelm et al., 2019), this aspect of ABC is perhaps unsurprising.

4.2 Fast Digit Style Isolation

Handwritten digits, such as from MNIST (LeCun & Cortes, 1998), have a natural separation of factors of variation into content and style. Here, content is the digit class (e.g., 2 or 8) and style is all remaining factors of variation (stroke width, slant, shape, etc.). Our goal is to generalize style across digit class, without grouped data where style is an inactive factor. Images are instead grouped by class into sets of size 64 and embedded to $\mathbb{R}^8$.

Figure 5a displays the first two principal components of learned embeddings of the digit 9, wholly unseen at training time and showing the invariance of the isolated style information to digit class. The instances fan out with regards to style factors of variation, most clearly the stroke thickness and slant. In Figure 5b we use test digits from each of the 10 classes to retrieve the most similar digits in other classes. We compare to the representations yielded by the discriminative approach of Sanchez et al. (2020) and CC-VAE (Jha et al., 2018), both of which learn a full description of the data which is partitioned into active and inactive parts by utilizing set supervision. Without having to learn a full description of the data, ABC yields style-correlated embeddings orders of magnitude faster than the related approaches.

4.3 Pose Transfer from Synthetic to Real Images

We showcase the full capabilities of ABC-X on the challenging task of object pose estimation. The goal is effective isolation of pose information which generalizes to the category level and across the synthetic to real domain gap, without any pose annotations for training. We leverage the freedom of one set per mini-batch to have additional active factors of variation to gradually incorporate wholly unsupervised real images.

We use the dataset included in KeypointNet (Suwajanakorn et al., 2018), which consists of renderings of ShapeNet (Chang et al., 2015) models from viewpoints randomly distributed over the upper hemisphere. Images are grouped according to their source 3D model (as in set $\mathcal{A}$ of Fig. 1) providing the set supervision for ABC-X. Other factors of variation such as object texture and lighting are also fixed, making orientation the only active factor within each set. We incorporate real images from the CompCars (Yang et al., 2015) and Cars196 (Krause et al., 2013) datasets for the car category, and 1000 images from the Pascal3d+ (Xiang et al., 2014) training split for chairs. Images are tight cropped and resized to 128x128.

The double augmentation loss (Section 3) helps bridge the domain gap by removing additional nuisance factors of variation which could shortcut the task of finding correspondence. Each image is randomly augmented twice with a combination of cropping, recoloring, and replacing the background with random crops from images of ImageNet-A (Hendrycks et al., 2019), following many of the augmentations used to bridge the synthetic to real domain gap in Sundermeyer et al. (2018; 2020). Each image is embedded to $\mathbb{R}^{64}$, using a network which places a few layers on top of an ImageNet-pre-trained ResNet50 (He et al., 2015). We found that cosine similarity with temperature $\tau = 0.1$ outperformed negative Euclidean distance.

In the first experiment, we train with set $\mathcal{A}$ purely synthetic and grouped by ShapeNet model, and $\mathcal{B}$ unconstrained. Set $\mathcal{B}$ gradually incorporates real images, ramping linearly to an average of 10% per set by the end of training.

	Cars		Chairs	
	Med Err (°) ↓	Acc@30° ↑	Med Err (°) ↓	Acc@30° ↑
ResNet (pre-trained)	16.1	0.66	43.9	0.41
CCVAE (Jha 2018)	54.8	0.26	79.5	0.19
ML-VAE (Bouchacourt 2018)	75.6	0.27	87.2	0.16
Set supervision w/ TCC loss (Dwibedi 2019)	22.1	0.57	63.9	0.38
Double augmentation only	85.4	0.33	80.0	0.20
ABC (no real images)	14.7	0.66	22.6	0.59
ABC-X (Full)	**12.8**	**0.72**	**15.6**	**0.75**

Table 1: ***Pose estimation without pose annotations at training.*** *Median error and accuracy on the Pascal3D+ car and chair test sets. Pose estimates are obtained through nearest neighbor lookup into 1800 synthetic images with associated GT pose; reported values are the average over ten randomly sampled codebooks. The full ABC-X method outperforms everything else.*

We evaluate on images from the Pascal3D+ test set by using nearest neighbor lookup with a codebook of 1800 synthetic images, unseen at training, with associated ground-truth pose (Table 1). We compare with the high-dimensional (D=16,384) output from the ResNet base network, and to embeddings learned with the VAE-based approaches of Jha et al. (2018) and Bouchacourt et al. (2018). The effects are most striking for the chair category, where category-level generalization and the domain gap are more difficult than for cars. The significant difference between ABC-X and the baseline approaches which learn full descriptions underscores the benefit of focusing on a partial description: with ABC-X the multitude of instance-related inactive factors need not be learned, and the incorporation of out-of-domain images with extra active factors is possible.

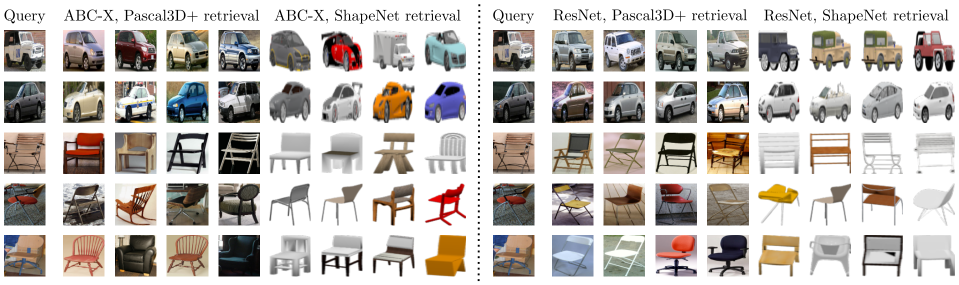

Query
ABC-X, Pascal3D+ retrieval
ABC-X, ShapeNet retrieval
Query
ResNet, Pascal3D+ retrieval
ResNet, ShapeNet retrieval

Figure 6: ***Retrieval results from ABC-X and ResNet embeddings.*** *Given a query image from the Pascal3D+ test split, we display the four nearest neighbors in embedding space, from the Pascal3D+ train split and from 1800 ShapeNet images. The accuracy and visual diversity of the ABC-X retrieval results illustrate effective isolation of pose information generalized across both the category and the synthetic-to-real domain gap.*

	Cars		Chairs	
	Med Err (°) ↓	Acc@30° ↑	Med Err (°) ↓	Acc@30° ↑
Liao et al. (2019)	12.3	0.85	30.8	0.49
+ ABC	11.0	0.79	28.1	0.52
+ ABC-X (2% unannotated real)	**9.3**	**0.87**	**26.0**	**0.55**

Table 2: ***Performance boost to spherical regression by incorporating ABC-X.*** *We show the effectiveness of incorporating ABC-X as an additional loss term when the data consists of annotated synthetic images and unannotated real images. ABC-X provides a means to incorporate the latter which helps bridge the domain gap.*

Ablative results illustrate the synergy of the components of ABC-X. Applying only the correspondence loss used by Dwibedi et al. (2019) in the limited setting of video alignment, we see reasonable performance on the car category but a failure to isolate pose in chairs. Excluding nuisance factors from the representations through augmentation, but without leveraging set supervision, does not yield pose-informative representations either. The incorporation of real images in ABC-X gives a sizable boost to performance over ABC, demonstrating the utility of the unsupervised data. Retrieval examples (Fig. 6) illustrate the generalization across instance and domain-specific factors of variation. Lookup results with the ABC-X representations are visually diverse and less erroneous in the synthetic-to-real jump than the high-dimensional ResNet embeddings.

In the second experiment (Table 2), we make use of pose annotations for the synthetic images by incorporating ABC-X into the spherical regression framework of (Liao et al., 2019), and operate in a scenario with fewer augmentations than in the first experiment. Specifically, we add a small spherical regression head on top of the ABC-X-conditioned representations and train on a weighted sum of the two losses. Even without any real images during training, ABC improves performance, presumably by better conditioning the intermediate latent space. A further boost to performance results with ABC-X when a small amount of real images (2%) are titrated in gradually over training, for both object categories. Thus ABC-X can be advantageous in scenarios where there is more supervision available than set supervision.

5 Discussion

The pursuit of bijective correspondence, essentially allowing the model to find its own positive pairs for use in a contrastive loss, offers a powerful new foothold into operating on factors of variation in learned representations. It is perfectly suited for domain transfer and the common real-world scenario where information about an abundance of unannotated real data is desired and related synthetic data is available.

ABC is significantly faster than related approaches (Fig. 5) because a full description of the data is not needed; indeed, not even all active factors of variation need be isolated (Figs. 3&4). The size of sets (Fig. 4d) and double augmentation (Table 1) grant ABC considerable control over the factors of variation which are isolated in the learned representations.

Ethics statement

We discuss a general form of representation learning, which is intentionally broad in its scope of potential applications. We have emphasized intuition and insight wherever possible in the aim to improve the accessibility of this and related research.

Reproducibility statement

Code to reproduce all results involving the Shapes3D and MNIST datasets from scratch are included as supplemental files and will be posted to a public repository upon manuscript acceptance. Implementation specifics for all experiments are included in Appendix G.

References

Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube, 2018.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 531–540. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/belghazi18a.html.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013. ISSN 0162-8828.

Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI*, February 2018.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020a. URL http://jmlr.org/papers/v21/20-163.html.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020b.

Taco S. Cohen and Max Welling. Transformation properties of learned visual representations. In *International Conference on Learning Representations (ICLR)*, 2015.

Enric Corona, Kaustav Kundu, and Sanja Fidler. Pose Estimation for Objects with Rotational Symmetry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7215–7222. IEEE, 2018.

Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pp. 4414–4423, 2017.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. *CoRR*, 2019.

Adar Elad, Doron Haviv, Yochai Blau, and Tomer Michaeli. Direct validation of the information bottleneck principle for deep nets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, Cambridge, MA, USA, 2004. MIT Press.

Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N. Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5548–5558, June 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230, 2018.

Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019. URL https://www.microsoft.com/en-us/research/publication/learning-deep-representations-by-mutual-information-estimation-and-maximization/.

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and V. S. R. Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders, 2018.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pp. 2539–2547, 2015.

Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Shuai Liao, Efstratios Gavves, and Cees G. M. Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Francesco Locatello, Stefan Bauer, Mario Lučić, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Frederic Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019.

Michael Mathieu, Junbo Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. Disentangling factors of variation in deep representations using adversarial training. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 5047–5055, 2016.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. *CoRR*, abs/1703.07464, 2017. URL http://arxiv.org/abs/1703.07464.

K L Navaneet, Ansu Mathew, Shashank Kashyap, Wei-Chih Hung, Varun Jampani, and R. Venkatesh Babu. From image collections to point clouds with self-supervised shape and pose networks, 2020.

Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi. Discovering relationships between object categories via universal canonical maps. In *CVPR*, 2021.

Brian Okorn, Mengyun Xu, Martial Hebert, and David Held. Learning Orientation Distributions for Object Pose Estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

Shaul Oron, Tali Dekel, Tianfan Xue, William T. Freeman, and Shai Avidan. Best-buddies similarity - robust template matching using mutual nearest neighbors, 2016.

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. *CoRR*, abs/1810.10510, 2018. URL http://arxiv.org/abs/1810.10510.

Eduardo Hugo Sanchez, Mathieu Serrurier, and Mathias Ortner. Learning disentangled representations via mutual information estimation. In *The European Conference on Computer Vision (ECCV)*, 2020.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, dec 2019. doi: 10.1088/1742-5468/ab3985. URL https://doi.org/10.1088%2F1742-5468%2Fab3985.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *Proceedings of International Conference in Robotics and Automation (ICRA)*, 2018. URL http://arxiv.org/abs/1704.06888.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgSwyBKvr.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. URL http://arxiv.org/abs/1703.05175.

Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O. Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9929–9939. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wang20k.html.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond PASCAL: A benchmark for 3d object detection in the wild. In *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 75–82, March 2014.

Lei Yang, Wenxi Liu, Zhiming Cui, Nenglun Chen, and Wenping Wang. Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes, 2020.

Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Qiang Zhang, Tete Xiao, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QIRlze3I6hX.

Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency, 2016.

Additional files submitted

The uploaded zip file contains an iPython notebook `abc_mnist.ipynb` with code to reproduce the MNIST digit style isolation results, a gif `abc_mnist_training_evolution.gif` showing the evolution of MNIST digit embeddings over the course of training with ABC, and a directory `shapes3d/` with code to reproduce the all results of Section 4.1 in the main text.

Appendix Contents

A Mutual information calculation, and corroboration with classification task

Computation of mutual information. To estimate the mutual information $I(U;G)$ for the Shapes3D experiments using MINE Belghazi et al. (2018), we train a statistics network T. We use a simple fully connected network whose input is the concatenation of the 64-dimensional embedding U and the 1-dimensional value for the particular generative factor G. It contains three layers of 128 units each with ReLU activations, with a final one-dimensional (scalar) output. The loss is the negated neural information measure of Belghazi et al. (2018),

$$\mathcal{L} = \log(\mathbb{E}_{u\sim P(U),g\sim P(G)}[\exp(T(u,g))]) - \mathbb{E}_{u,g\sim P(U,G)}[T(u,g)] \tag{1}$$

At a high level, the network exploits the difference between the joint distribution $P(U, G)$, where the embedding is properly matched with its correct generative factor, and the product of marginals $P(U)P(G)$, which is simulated by shuffling the labels for the first term in the loss. This difference between the joint and the marginals is the mutual information of the two variables. We train with a learning rate of 3×10^{-4} and a batch size of 256 for 20,000 steps, which we found to be sufficient for convergence. The estimate of the mutual information we report is the average value of the neural information measure over 256,000 samples from the dataset. A new statistics network is trained for each of the six generative factors.

To deal with the determinism of the embedding network, we add Gaussian distributed noise $\eta \sim \mathcal{N}(0, \sigma^2)$ directly to the embeddings. For the noise scale sweeps in Figure 3c,d we repeat the calculation for 40 logarithmically spaced values of σ.

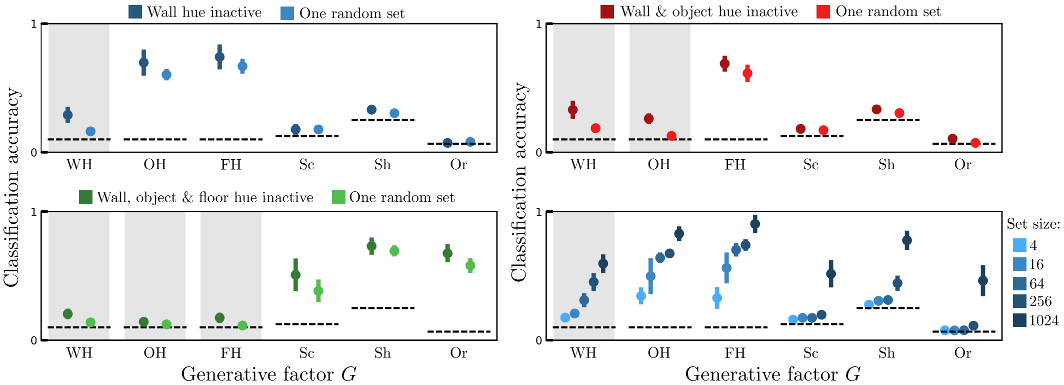


Figure 7: ***Corroborating I_{MINE} with classification task.*** *As a proxy for the mutual information, we use the test set classification accuracy of networks trained to predict the six generative factors, one network per factor. As before, the shaded columns indicate which of the generative factors were inactive while training ABC. Gaussian-distributed random noise with $\sigma = \sqrt{\tau}$ was added to the embeddings to effectively remove information on length scales less than the characteristic length scale of the ABC loss. The dashed lines show the classification accuracy that would result from random guessing.*

I_{MINE} versus classification accuracy. To corroborate the Shapes3D mutual information measurements of Section 4.1, we use the common approach of training a simple classifier which takes the learned representations as input and tries to predict the generative factors (Figure 7). We train a different classifier for each generative factor, and use an architecture of 3 fully connected layers with 32 units each, ReLU activation. As with the measurements of mutual information, there is the issue of evaluating a deterministic network which in general preserves all information Elad et al. (2019). By adding Gaussian noise with magnitude $\sigma = \sqrt{\tau}$, the classification task qualitatively reproduces the behavior of Figure 4. Namely, when one or two hue factors are inactive, information about the remaining hue factor(s) is enhanced and information about the inactive factor(s) is suppressed. When all three hue factors are inactive, then and only then is information about the three geometric factors enhanced. There is no substantial difference in the semi-supervised setting, where one set of each mini-batch has no inactive factors.

B The role of length scales in isolating factors of variation

The ABC loss operates over a characteristic scale in embedding space, set by the temperature parameter τ which plays a role in both the soft nearest neighbor calculation and the InfoNCE loss. When using a similarity measure derived from Euclidean distance, this characteristic scale may be interpreted as a length scale.

Two embeddings which are separated by less than this length scale effectively have a separation of zero in the eyes of the loss, and there is no incentive to further collapse them.

To be specific, when using L2 (Euclidean) distance as the similarity metric, the temperature τ is the characteristic length scale. When using L2 squared distance, as in the MNIST and

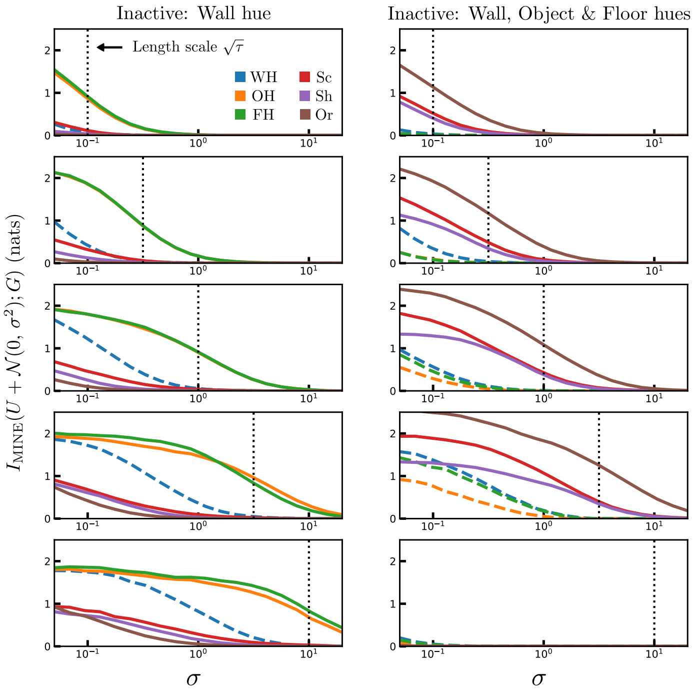


Figure 8: ***Temperature sets the length scale of the cutoff between active and inactive factors.*** *We train with negative squared Euclidean distance between embeddings as the similarity measure, which makes $\sqrt{\tau}$ a natural length scale for embedding space. By varying the temperature used during training (varying vertically), we mark the length scale $\sqrt{\tau}$ with a dotted vertical line in each subplot. Predictably, the magnitude of the noise σ at which inactive factors are suppressed scales with $\sqrt{\tau}$. Had negative Euclidean distance been used instead, we would expect the scaling to follow τ. The bottom right subplot shows one of the limits of varying the temperature of the ABC loss: when it is too large compared to the spread of the initialized embeddings, training is often unsuccessful.*

Shapes3D experiments, the square root of the temperature is the characteristic length scale. With cosine similarity, as in the pose estimation experiments of Section 4.3, temperature sets a characteristic angular difference between embeddings. It is less straightforward to probe by these information measurements, and irrelevant for actually performing lookup or regression.

For downstream tasks, including lookup using the embeddings, this length scale is generally irrelevant. However, measuring the mutual information requires the addition of noise with a particular scale, and the freedom in choosing this parameter begs the question of a relevant scale in embedding space. As a fortunate consequence, it allows a precise definition of the factor isolation that results from ABC. We show in Figure 8 several Shapes3D experiments where the temperature τ during training took different values. The mutual information is measured as in Figure 3c,d with a sweep over the magnitude of the added noise.

The vertical dashed line in each run shows the characteristic length scale, $\sqrt{\tau}$, and it is clear to see information about the inactive factor(s) (indicated by dashed lines) decaying to zero below the length scale. The predicted behavior, of object and floor hue being isolated when wall hue is inactive, and of the geometric factors being isolated when all three hue factors are inactive, happens in nearly all the runs. The length scales of everything, as measured by the magnitude σ of the noise where the information decays, expand with increased temperature.

There is a limit to this behavior, however, which is shown in the bottom right subplot. When the temperature is too large compared to the initial separations of the embeddings, there is too little gradient information for even the Adam optimizer to leverage, and training is unsuccessful.

Summary. ABC's isolation of factors is a matter of scales in embedding space, and this allows the method to be well-suited for lookup tasks. Information about inactive factors is confined to scales less than the characteristic scale set by the temperature during training, and the isolated active factors inform the structure of embedding space over larger scales.

C Why does ABC isolate multiple factors of variation instead of a single one?

If correspondence between two sets can be found with only a single factor of variation common to both sets, why do the experiments of this paper suggest ABC isolates multiple factors of variation? To be specific, in almost all of the Shapes3D experiments, multiple generative factors were present in the learned representations. Presumably a one-to-one correspondence between MNIST digits could be found using stroke thickness alone, yet the embeddings almost always contained slant information as well. In the pose experiments, only embedding azimuth would suffice to allow a correspondence between images, yet elevation information was also present.

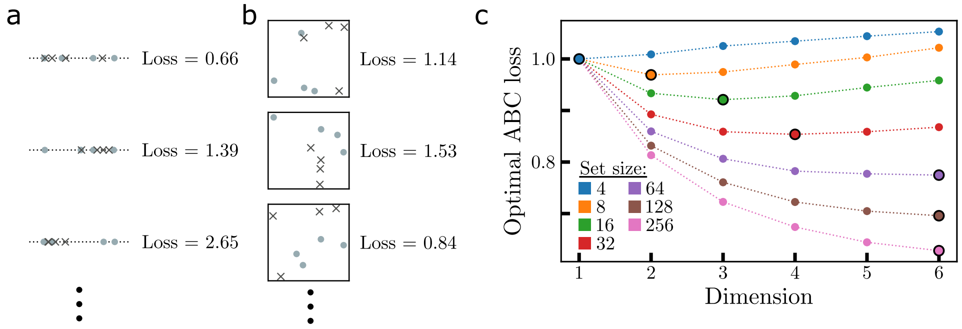


Figure 9: ***The case for finding more than one factor of variation, through a simple example.*** *We model the embeddings that would be learned from randomly distributed factors of variation as points sampled uniformly over the unit interval in one to six dimensions.* ***(a)*** *Displayed are three random draws, with set size 4 and dimension 1, and corresponding ABC loss values. The × and circle markers designate randomly generated set $\mathcal{U}$ and set $\mathcal{V}$.* ***(b)*** *Same as* ***(a)****, but for 2 dimensions.* ***(c)*** *The ABC loss averaged over 10,000 pairs of random sets, by set size and the dimension of the embedding distribution. The dimension of the embedding distribution serves as a model for the number of independent factors of variation which are isolated. As the set size grows, the dimension which yields the minimal loss (outlined markers) grows as well.*

In Figure 9 we run a simple Monte Carlo experiment where embeddings are simulated by randomly sampling from a uniform distribution over the hypercube in different dimensions. This represents the ideal case at the end of training with ABC, where all sets of embeddings are distributed identically. For a given set size, we vary the dimension of the embedding space as an analogue for the number of independent factors isolated. In this simplified setting, we are able to exclude any effects of the salience of different factors, and focus only on the value of the loss as stochastic embeddings are spread over different numbers of dimensions.

The ABC loss is averaged over 10,000 random draws, and we rescale by the loss in one dimension. In the normal training setting the distribution can adapt to the temperature τ (Section B). In this simulation, the distribution is fixed so the temperature which optimizes the loss needs to be found; we numerically optimize it.

In Figure 9c we find non-monotonic dependence of the loss on the dimension, suggesting competing influences on the loss. Additionally, the dimension which minimizes the loss for a given set size grows with the set size. Using the number of dimensions as a model for the number of independent factors of variation isolated in the embeddings, this and the increasing magnitude of the slope as the set size increases show increasing pressure to embed with respect to more factors as the set size grows. This matches the behavior of Figures 4d and 7, where training ABC with the largest set size led to the isolation of all five active factors, and with the smallest set size to the partial isolation of only two.

D Ablative studies on the pose estimation tasks

In Figures 10 and 11 we show ablative studies on the pose estimation experiments of Section 4.3, for training with the ABC loss where pose is ultimately extracted using a lookup table (Table 1) and the experiment where the ABC loss combined with the spherical regression method of Liao et al. (2019) (Table 2).

On both tasks, there is an optimal proportion of real images, though it is much lower for regression. Gradual titration of real images into the unconstrained set $\mathcal{B}$ was neutral or negative for the lookup task (Figure 10, top row) and generally positive for the regression task (Figure 11, top row). Cosine similarity outperforms negative Euclidean distance, and we show the dependence on temperature τ in the second row of Figure 10.

The car and chair categories present different challenges for pose estimation – e.g. an approximate front-back symmetry for cars, greater class diversity for chairs, outdoor versus indoor settings for cars versus chairs, etc. Several of the ablated factors cause differing effects on the performance for the two categories.

For instance, there is an apparent difference between the two categories in the dependence on the augmentation scheme, shown in the third row of Figure 10. Randomly translating the bounding box by 0.1 of its height and width helps both categories, but more than that and the chair performance greatly suffers.

Another difference between the categories is seen in the final row of Figure 10, where increasing the set size during training only helps pose estimation on cars. For the largest set size, however, chair pose estimation begins to suffer. We presume the pressure to isolate more active factors of variation from increased set size, discussed in Section C, can actually be harmful to the pose estimation task if unrelated factors confound the pose estimation during lookup. Set size similarly shows mixed effects for the regression task, shown in the final row of Figure 11.

E Augmentations used for pose estimation

For each real and synthetic image in the pose estimation tasks of Section 4.3, we augment twice and train with the double augmentation version of the ABC loss, in order to suppress additional nuisance factors from the learned representations. We show in Figure 12 sample augmentation of real and synthetic car images, which include random translations of the bounding box, brightness adjustment, the addition of salt and pepper noise to each pixel, the addition of a scaled, Sobel-filtered version of the image, and hue adjustment for the real images. We also paint the background of the synthetic images with random crops from ImageNet-A Hendrycks et al. (2019).

F Extended digit style isolation results

In Figure 13 we compare digit style isolation on MNIST using the output of ABC and the style part of the latent representations yielded by the VAE-based approaches of Jha et al. (2018) and Bouchacourt et al. (2018). Interestingly, ML-VAE appears to embed the digits with respect to stroke thickness and slant very similarly to ABC at the beginning of training, long before any realistic images are able to be generated, but this clear interpretability of the embeddings fades as training progresses.

G Hyperparameters and implementation details

For all experiments we use the ADAM optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). Padding for convolutional layers is always ‘valid.’

G.1 Shapes3D

For the experiments of Figures 3&4 we used the network architecture listed in Table 3, and trained for 2000 steps with a learning rate of 3×10^{-5}. We used a stack size of 32 and

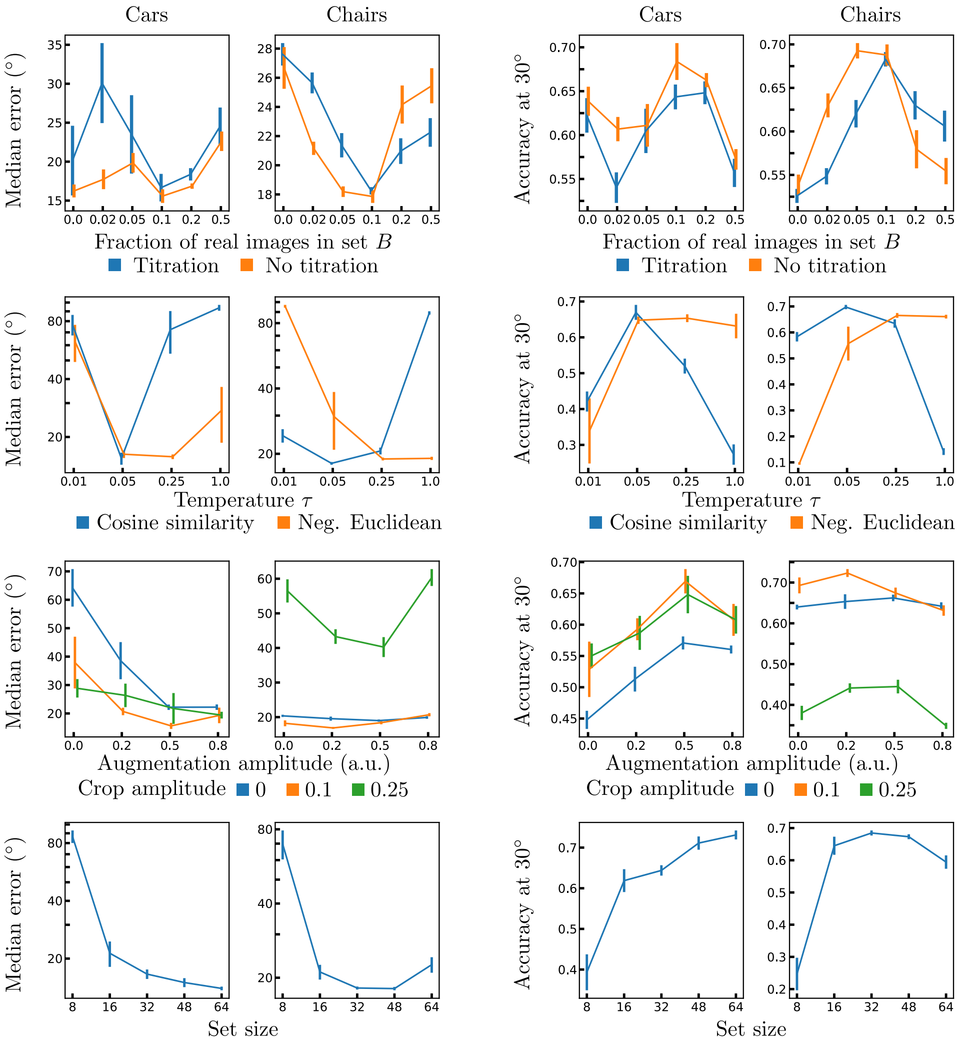


Figure 10: ***Ablative studies on Pascal3D+ pose lookup with ABC embeddings.*** *Error bars are the standard error of the mean over 8 random seeds for each configuration. We show results on the Pascal3D+ test split for the car and chair categories. For each row, the training configuration is the same as described in Section G with only the listed aspect of training being changed. In the first row, no titration means to the fraction of real images in set B are present from the beginning of training. The augmentation amplitude in the third row controls the coloring changes discussed in Section E. The crop amplitude is another form of augmentation, though we separate it for clarity. It controls the random translation of the bounding box, as a fraction of the dimensions of the bounding box.*

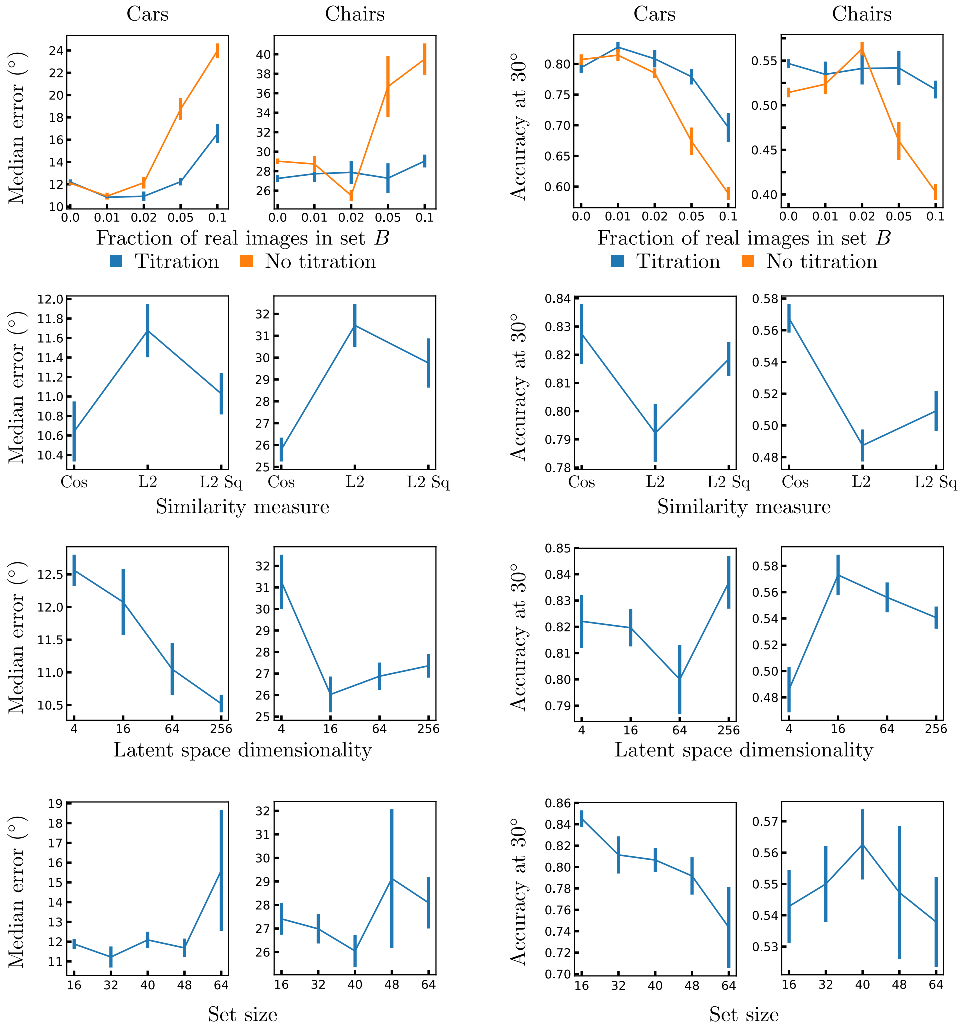


Figure 11: ***Ablative studies on Pascal3D+ with spherical regression + ABC network.*** *Error bars are the standard error of the mean over 10 random seeds for each configuration, with less than 1% of the runs discarded for lack of convergence. We show results on the Pascal3D+ test split for the car and chair categories. For each row, the training configuration is the same as described in Appendix G with only the listed aspect of training being changed. In the first row, no titration means to the fraction of real images in set B are present from the beginning of training. The three similarity measures in the second row are cosine similarity, L2 (Euclidean) distance, and squared L2 distance.*

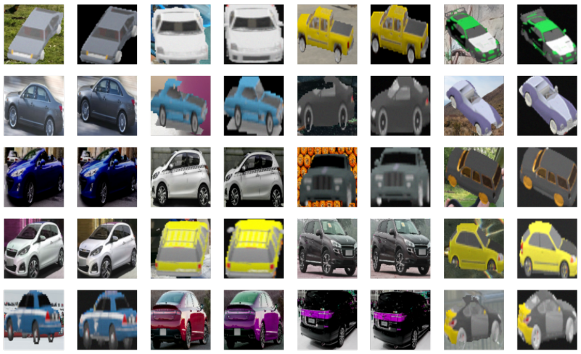


Figure 12: ***Augmentations used in the pose estimation experiments.*** *We show sample augmentations applied to both real and synthetic cars. These include adjusting brightness and hue, adding normally distributed noise to each pixel, random translations of the crop (bounding box), and replacing the background of synthetic images with random crops from real images.*

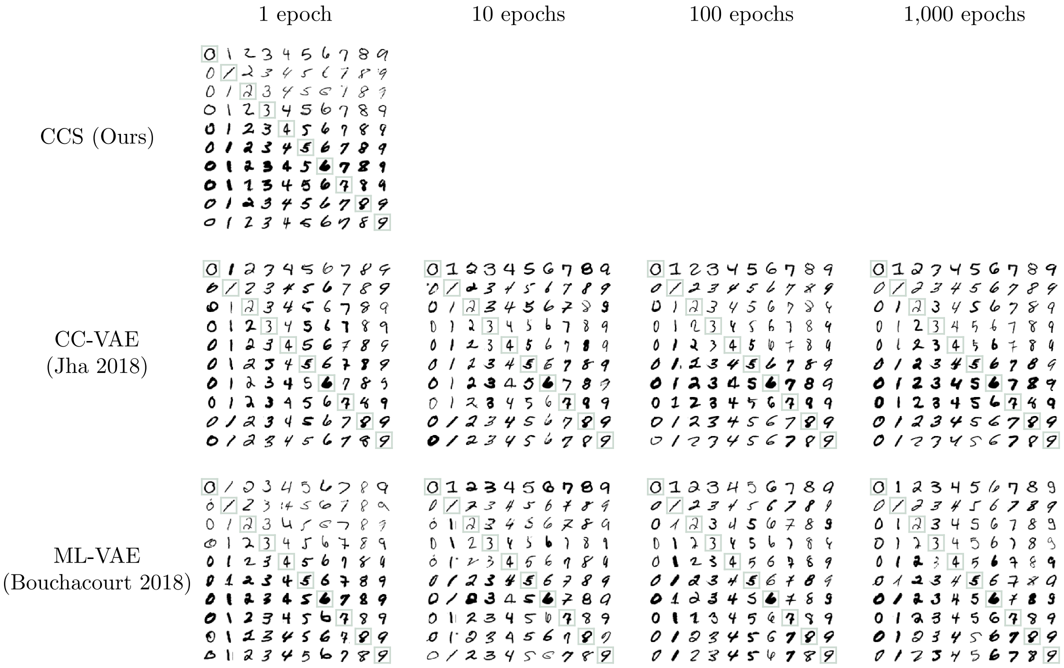


Figure 13: ***Retrieval results over the course of training, comparison.*** *We compare retrieval on the test set of MNIST at various stages of training ABC and the two VAE-based approaches mentioned in the main text. As in Figure 5, the query images are the boxed images along the diagonal, and each row is the nearest representative for each class in embedding space. Also as before, in all cases the digit 9 was withheld during training.*

squared L2 distance as the embedding space metric, with a temperature of 1. To curate a set for training, we randomly sample from among the possible values for the inactive factor(s) and then filter the dataset according to it. This takes longer when there are more inactive factors, as more of the dataset must be sieved out to acquire each stack.

G.2 MNIST

For the MNIST experiments we used the architecture specified in Table 4. The stack size was 64. We used a learning rate of 10^{-4} and trained for 500 steps. We used squared L2 distance as the embedding space metric and a temperature of 1. All instances of the digit 9

Layer	Units	Kernel size	Activation	Stride
Conv2D	32	3x3	ReLU	1
Conv2D	32	3x3	ReLU	1
Conv2D	64	3x3	ReLU	2
Conv2D	64	3x3	ReLU	1
Conv2D	128	3x3	ReLU	1
Conv2D	128	3x3	ReLU	2
Flatten	–	–	–	–
Dense	128	–	ReLU	–
Dense	Embedding dimension (64)	–	Linear	–

Table 3: **Architecture used for Shapes3D experiments (Section 4.1).** Input shape is [64, 64, 3].

Layer	Units	Kernel size	Activation	Stride
Conv2D	32	3x3	ReLU	1
Conv2D	32	3x3	ReLU	1
Conv2D	32	3x3	ReLU	2
Conv2D	32	3x3	ReLU	1
Conv2D	32	3x3	ReLU	1
Flatten	–	–	–	–
Dense	128	–	ReLU	–
Dense	Embedding dimension (8)	–	Linear	–

Table 4: **Architecture used for MNIST experiments (Section 4.2).** Input shape is [28, 28, 1].

are held out at training time, and images of the other digits are formed into stacks before being randomly paired each training batch. This ran in under 30 seconds on an NVIDIA Tesla V100 GPU.

G.3 Pose estimation

Layer	Units	Kernel size	Activation	Stride
ResNet50, up to conv4_block6	–	–	–	–
Conv2D	256	3x3	ReLU	1
Global Average Pooling	–	–	–	–
Flatten	–	–	–	–
Dense	128	–	tanh	–
Dense	Embedding dimension (64)	–	Linear	–

Table 5: **Architecture used for pose estimation experiments (Section 4.3).** Input shape is [128, 128, 3].

For both the pose estimation lookup (Table 1) and regression (Table 2) tasks, we use the same base network to embed the images, described in Table 5. In contrast to the Shapes3D and MNIST experiments, we train with mini-batches consisting of 4 pairs of image sets, each of size 32. We use cosine similarity and a temperature of 0.1 for lookup and 0.05 for regression. For the lookup task, the network trained for 40k steps with a learning rate that starts at 10^{-4} and decays by a factor of 2 every 10k steps. The beginning of training is purely synthetic images and then ramping up linearly to 10% real images folded into the unconstrained stack, stepping every 4k steps.

For regression, the embeddings are then fed, separately for each Euler angle, as input to a 128 unit dense layer with tanh activation, which is then split off into two dense layers with 2 and 4 units and linear activation for the angle magnitude and quadrant, respectively, as in (Liao et al., 2019). To maintain consistency between how the embeddings are processed for the ABC loss and how they are fed into the regression sub-network, the embeddings are L2-normalized to lie on the 64-dimensional unit sphere before the regression. The angle magnitudes are passed through a spherical exponential activation function Liao et al. (2019), which is the square root of a softmax. The magnitudes are then compared with ground truth $(|\sin\phi_i|, |\cos\phi_i)|$, with i spanning the three Euler angles, through a cosine similarity loss. The quadrant outputs are trained as a classification task with categorical cross entropy against the ground truth angle quadrants, defined as $(\text{sign}(\sin\phi_i), \text{sign}(\cos\phi_i))$. Training proceeds for 60k steps with a learning rate that starts at 10^{-4} and decays by a factor of 2 every 20k steps.

To more closely match the distribution of camera pose in real images, we filter the ShapeNet renderings by elevation: 0.5 radians and 1.3 radians for the max elevation for cars and chairs, respectively.

G.4 Baselines

Imagenet-pretrained ResNet: We use the same ResNet50V2 base as for the ABC embedding network, and compare representations for each image by cosine similarity (which performed better than comparing by L2 distance).

Sanchez et al. (2020): We used the colored-MNIST architecture specifications and hyperparameters described in the Supplemental Material for the MNIST experiments of Section 4.2. As the colored-MNIST factors of variation isolated by Sanchez et al. (2020) are simpler in nature (color of foreground/background from specific digit, versus digit identity from style), we found better results by boosting the dimension of the exclusive representation to 64 (up from the original 8 for the color description).

We replicated the architecture and hyperparameters used in the Shapes3D experiments by Sanchez et al. (2020) for the pose lookup experiments, downsizing the ShapeNet renderings and Pascal3D+ tight crops to 64x64 RGB images to match the architecture used.

Jha et al. (2018) and Bouchacourt et al. (2018): We translated the publicly available pytorch code to tensorflow for training MNIST [1], [2]. We were unable to find code for their experiments on larger image sizes, but we followed the encoder and decoder specifications for the 64x64 RGB images in the Supplemental for Jha et al. (2018), found here[3], for both methods. We optimized hyperparameters in a grid search around the published numbers, and used a group size for Bouchacourt et al. (2018) which matched the stack size used for the ABC method. As with Sanchez et al. (2020), we downsized the ShapeNet renderings and Pascal3D+ tight crops to 64x64, after attempts to scale the encoder-decoder architecture up to 128x128 were unsuccessful.

[1] https://github.com/ananyahjha93/cycle-consistent-vae

[2] https://github.com/DianeBouchacourt/multi-level-vae

[3] https://arxiv.org/pdf/1804.10469.pdf