A. Recent Descriminative SSL Formulations

Notation	Description
$\mathbf{x}_i$	An image sample
$\mathcal{A}_1(\mathbf{x}_*), \mathcal{A}_2(\mathbf{x}_*)$	Two random augmentations
$f_\theta(.)$	A feature encoder parameterized by θ
$f_\xi(.)$	An EMA encoder parameterized by ξ
τ	A temperature scaling term
$p_\theta(.)$	An MLP predictor parameterized by θ.
`sg`	`stop-gradient` operation.

Table 3: Notations for important elements in SSL. **Note that the extracted or projected representations are normalized to a unit sphere unless specified otherwise.**

SimCLR[10] optimizes InfoNCE to maximize similarity between positive pairs and minimize similarity between negative pairs. Positive pairs are two augmented views of an image sample, $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. Negative pairs are all other augmented samples in a mini training batch. For a batch with N image samples, the augmentation produces $2N$ augmented samples. A feature encoder $f_\theta(.)$ extracts representations of the batch data $\tilde{\mathbf{z}}_i = f_\theta(\tilde{\mathbf{x}}_i)$. SimCLR optimizes the following objective:

$$\mathcal{L}_{SimCLR} = \mathop{\mathbb{E}}_i[-log \frac{e^{\tilde{\mathbf{z}}_{2i}^\top \tilde{\mathbf{z}}_{2i+1}/\tau}}{\sum_{j=1, j\neq 2i}^{2B} e^{\tilde{\mathbf{z}}_{2i}^\top \tilde{\mathbf{z}}_j/\tau}}]$$

MoCo/MoCo V2 optimzes InfoNCE to maximize similarity between positive pairs and minimize similarity between negative pairs. Positive pairs are two augmented views of an image sample, $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. But negative pairs are representations learned via a moving-averaged network, $f_\xi(.)$, and stored in a memory bank with size K, $\mathcal{M}_K$. And $\xi = m\xi + (1-m)\theta$, where m is a momentum coefficient. $\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$, $\hat{\mathbf{z}}_{2i+1} = f_\xi(\tilde{\mathbf{x}}_{2i+1})$, $\hat{\mathbf{z}}_j = f_\xi(\tilde{\mathbf{x}}_j) \in \mathcal{M}_K$. MoCo optimizes the following objective:

$$\mathcal{L}_{MoCo} = \mathop{\mathbb{E}}_i[-log \frac{e^{\tilde{\mathbf{z}}_{2i}^\top \hat{\mathbf{z}}_{2i+1}/\tau}}{\sum_{j=1}^{K} e^{\tilde{\mathbf{z}}_{2i}^\top \hat{\mathbf{z}}_j/\tau}}]$$

BYOL aligns the projection of a representation of an augmented sample with an EMA representation of another augmented sample. The main difference between BYOL and SimCLR/MoCo is the claim that BYOL only formulates the objective on positive pairs. $\tilde{\mathbf{x}}_{2i}, \tilde{\mathbf{x}}_{2i+1} = \mathcal{A}_1(\mathbf{x}_i), \mathcal{A}_2(\mathbf{x}_i)$. An MLP predictor, $p_\theta(.)$, further projects the representation extracted by the feature encoder to an embedding space and the *EMA* representation predicts the projected embedding/representation by alignment. $\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$,$\hat{\mathbf{z}}_{2i+1} = f_\xi(\tilde{\mathbf{x}}_{2i+1})$. BYOL optimizes the following objective:

$$\mathcal{L}_{BYOL} = \mathop{\mathbb{E}}_i \|p_\theta(\tilde{\mathbf{z}}_{2i}) - \hat{\mathbf{z}}_{2i+1}\|_2^2$$

SimSiam aligns the projection of a representation of an augmented sample with a detached representation of another augmented sample. Unlike BYOL or MoCo, SiamSiam omits the EMA encoder that the author deem to be unnecessary for a stable representation learning. An MLP predictor, $p_\theta(.)$, further projects the representation extracted by the feature encoder to an embedding space and the *detached* representation predicts the projected embedding/representation by alignment. $\tilde{\mathbf{z}}_{2i} = f_\theta(\tilde{\mathbf{x}}_{2i})$,$\hat{\mathbf{z}}_{2i+1} = f_\theta(\tilde{\mathbf{x}}_{2i+1})$[4]. SimSiam optimizes the following objective:

$$\mathcal{L}_{SimSiam} = \mathop{\mathbb{E}}_i[-\frac{p_\theta(\tilde{\mathbf{z}}_{2i})}{\|p_\theta(\tilde{\mathbf{z}}_{2i})\|_2} \cdot \frac{\hat{\mathbf{z}}_{2i+1}}{\|\hat{\mathbf{z}}_{2i+1}\|_2}]$$

Barlow Twins aligns positive representations of two augmented samples in feature dimensions and reduces redundancy cross different feature dimensions. Different to all aforementioned SSL methods, the authors suggest to standard normalize the representation (zero mean and unit std) instead of unit sphere normalization. However, as stated in the paper, either normalization scheme works under Barlow Twin method. $\mathcal{Z}^A = \{\tilde{\mathbf{z}}_{2i}\}_{i=1}^N = \{f_\theta(\tilde{\mathbf{x}}_{2i})\}_{i=1}^N$,$\mathcal{Z}^B = \{\tilde{\mathbf{z}}_{2i+1}\}_{i=1}^N = \{f_\theta(\tilde{\mathbf{x}}_{2i+1})\}_{i=1}^N$. And $\mathcal{Z}^A$ and $\mathcal{Z}^B$ are normalized over the batch statistics. Barlow Twins optimizes the following objective:

$$\mathcal{L}_{BarlowTwins} = \sum_a (1 - \mathcal{C}_{aa})^2 + \lambda \sum_a \sum_{b\neq a} {\mathcal{C}_{ab}}^2 \tag{13}$$

$$\mathcal{C}_{ab} = \frac{\sum_{i=1}^N [\tilde{\mathbf{z}}_{2i}]_a [\tilde{\mathbf{z}}_{2i+1}]_b}{\sqrt{\sum_i ([\tilde{\mathbf{z}}_{2i}]_a)^2} \sqrt{\sum_i ([\tilde{\mathbf{z}}_{2i+1}]_b)^2}}$$

B. Extended Theory and Proofs

In sections 3.1 and 3.2 we introduce our data generation process and prove that all SSL methods benefit from the alignment term in their objectives. Here we extend the theory to include the output entropy(s) of the encoder network(s) and provide analysis on how SSL prevent representation collapse by maximizing the output entropy of the network.

Theorem B.1 *With a data generation process described in 3.1, all discriminative SSL objectives have an alignment loss function between positive pairs from the network and output entropy loss function(s) of the network(s):*

$$\mathcal{L}_{SSL} = \|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2 - H(f(\mathbf{x}, \theta)) \tag{14}$$

[4]The $\mathbf{z}$ is not normalized yet, since in the loss function both projected and extracted representations are normalized.

For InfoNCE-driven SSL methods, the proof is:

$$\lim_{K\to\infty}\mathcal{L}_{InfoNCE} - logK = \underbrace{-\frac{1}{\tau}\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[f(\mathbf{x})^\top f(\tilde{\mathbf{x}})]}_{alignment} + \underbrace{\mathop{\mathbb{E}}_{\mathbf{x}}[log \mathop{\mathbb{E}}_{\mathbf{x}^-}[e^{f(\mathbf{x}^-)^\top f(\mathbf{x})/\tau}]]}_{uniformity} \quad (15)$$

with the *alignment* term in (15) equivlent to $\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}(1 - \|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2/2)$ and the *uniformity* term equivalent to $-H(f(\mathbf{x})) + logC_q(\mathbf{z})$ [1]. Hence complete the proof by:

$$\lim_{K\to\infty}\mathcal{L}_{InfoNCE} - logK = \frac{1}{\tau}\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[\|f(\mathbf{x}) - f(\tilde{\mathbf{x}})\|_2^2/2 - 1] - H(f(\mathbf{x})) + logC_q(\mathbf{z}) \quad (16)$$

For both EMA-driven and Siamese with predictor SSLs, we show that the loss function can be reformulated to three terms that first two are the alignment between positive pairs through the online/trainable network, and the alignment between same data sample from two networks. The third term can be further approximated via second order Taylor expansion around $\mathbf{z}$.

$$\begin{aligned}
\mathcal{L} &= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}\|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta) + p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}\|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}\|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top(p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi))] \\
&\approx \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}\|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}\|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 + 2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}]) + 2log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}[e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi)}]) \\
&\quad -2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi)}]) \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}\|p'(\mathbf{x},\theta) - p'(\tilde{\mathbf{x}},\theta)\|_2^2 + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}\|p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi)\|_2^2 \\
&\quad -2 - 2H(p'(\mathbf{x})) + 2logC_q(\mathbf{z}) + 2log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}[e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi)}]) \\
&\quad -2log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi)}])
\end{aligned} \quad (18)$$

$$\begin{aligned}
&-2\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[(p'(\tilde{\mathbf{x}},\theta) - p'(\mathbf{x},\theta))^\top(p'(\tilde{\mathbf{x}},\theta) - f(\tilde{\mathbf{x}},\xi))] \\
&= -2(1 - \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)] - \\
&\quad \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}[p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))] + \mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]) \\
&\approx -2(1 - log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}]) + \frac{\mathbb{V}[p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)]}{2\mathbb{E}[p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)]^2} \\
&\quad -log(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}}}[e^{p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi)}]) + \frac{\mathbb{V}[p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))]}{2\mathbb{E}[p'(\tilde{\mathbf{x}},\theta)^\top f(\tilde{\mathbf{x}},\xi))]^2} \\
&\quad +log(\mathop{\mathbb{E}}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi)}]) - \frac{\mathbb{V}[p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]}{2\mathbb{E}[p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi))]^2})
\end{aligned} \quad (17)$$

Assuming that κ_2 is a large number (as set by $1/\tau$ in SimCLR and MoCo), then the variance terms $\mathbb{V}[.] \approx 0$.

Since $f(\mathbf{x},\theta)$ and $f(\tilde{\mathbf{x}},\xi)$ maps in the same space $\mathbb{R}^{d_2}$, p can be considered as a bijective linear transformation within $\mathbb{R}^{b_2}$. In [22] by change of variable: $H(\mathbf{Y}) = H(\mathbf{X}) + \mathbb{E}[log|\mathbf{J}_m|] - H(\mathbf{X}|\mathbf{Y})$ if $\mathbf{Y} = m\mathbf{X}$, where m is a projection matrix mapping $\mathbf{X} \to \mathbf{Y}$ and $\mathbf{J}_m$ is the Jacobian of m, $\frac{\delta m}{\delta \mathbf{x}}$. This relates the last two terms in (18) to maximizing the output cross entropy of p' and f w.r.t the same sample, and minimizing the output cross entropy of p' and f w.r.t positive samples. This also hints on the importance of the predictor in BYOL, since removing the p in $p \circ f$, $2log(\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top p'(\tilde{\mathbf{x}},\theta)}])$ and $-2log(\mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}})}[e^{p'(\mathbf{x},\theta)^\top f(\tilde{\mathbf{x}},\xi)}])$ will cancel out and omit the output entropy maximization objective resulting in the representation collapse.

The same analysis applies to Siamese with predictor SSL. In case the predictor and/or `stop-gradient` is removed, the output entropy maximization objective will be no longer available and lead to a trivial solution.

For Barlow Twins, the objective can be regarded as minimizing the information loss between two augmented examples. In Appendix.A in [74], the author formulate such relation to the Information Bottleneck Principle:

$$\begin{aligned}
\mathcal{IB}_\theta &= I(f_\theta(\mathbf{x}),\mathbf{x}) - \beta I(f_\theta(\mathbf{x}),\tilde{\mathbf{x}}) \\
&= H(f_\theta(\mathbf{x})|\mathbf{x}) + \frac{1-\beta}{\beta}H(f_\theta(\mathbf{x}))
\end{aligned} \quad (19)$$

The first term in (19) is linked to the alignment term in (13) when [74] assumes that $f(\mathbf{x})$ follows a Gaussian distribution. However, in our representation learning formulation, we assume the conditional distribution of positive pairs follows a vMF distribution (3). We can further decompose

(19):

$$\mathcal{IB}_\theta = \mathcal{L}_{alignment} - H(f_\theta(\mathbf{x})) + \frac{1-\beta}{\beta} H(f_\theta(\mathbf{x})) \quad (20)$$

As suggested in [74], when $\beta < 1$ the best solution of (20) is to set the representation to a constant, i.e. representation collapse. When $\beta > 1$, the last term in (20) is the same as maximizing the output entropy of the network.

Once we prove that all SSL objectives contain the alignment term and output entropy maximization term, we demonstrate that the cross entropy between $p(.|\mathbf{z})$ in (1) and $q_h(.|\mathbf{z})$ in (3) can be formulated with the $\mathcal{L}_{alignmet} - H(f_\theta(\mathbf{x}))$ as illustrated in [76].

$$\begin{aligned} &\mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[H(p(.|\mathbf{z}), q_h(.|\mathbf{z}))] \\ &= -\mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}\sim p(\tilde{\mathbf{z}}|\mathbf{z})}[log(q_h(\tilde{\mathbf{z}}|\mathbf{z}))]] \\ &= \kappa_2 \mathop{\mathbb{E}}_{(\mathbf{z},\tilde{\mathbf{z}})}[h(\tilde{\mathbf{z}})^\top h(\mathbf{z})] - \mathop{\mathbb{E}}_{\mathbf{z}}[log(\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}}[e^{h(\tilde{\mathbf{z}})^\top h(\mathbf{z})\kappa_1}])] \\ &= \mathcal{L}_{alignment} - H(h(\mathbf{z})) \end{aligned} \quad (21)$$

Since $H(.) \leq 0$, then the alignment loss will be a lower bound for $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[H(p(.|\mathbf{z}), q_h(.|\mathbf{z}))]$.

Finally, we can use **Proposition 1** and **Proposition 2** in [76] to prove 3.2 and 3.3.

C. Causal3DIdent

3DIdent contained 7 object classes: Teapot, Hare, Dragon, Cow, Armadillo, Horse, Head. For spotlight position, spotlight hue, and background hue, variable values are sampled from $U(-1, 1)$. The dependence is imposed by varying the mean (μ) of a truncated normal distribution with standard deviation $\sigma = 0.5$, truncated to the range $[-1, 1]$. See Appendix B in [67] for the dependency on μ.

To exclude variable range with extreme values, we exclude edges for uniformly sampled variables (≤ -0.8 and ≥ 0.8), and exclude smaller portion tail for dependent variables ($\leq \mu - 0.8$ if $\mu > 0$,$\geq \mu + 0.8$ if $\mu \geq 0$). To ensure the training data size is the same after sampling, the included samples are delicately sampled to match the original data size. Three illustrative examples are shown in Figure 9.

This section contains the values of hyperparameters for SSL methods. **Note the batch size is set to 128**.

To verify that the difference between seen and unseen distribution does not induce large discrepancy in the performance, we evaluate the accuracy on all classes and report the difference between seen and unseen distributions. See Table 5 for results. On average, the difference is accuracy is about 2% to 3%, which is not comparable to the minimum reduction in accuracy $\approx 20\%$ reported in Figure 4.

We also verify the same deterioration in performance when only intervening the 5 children nodes in Figure 3. In Figure 10, we observe the same deterioration when only 5 variables are sampled to exclude the extreme edge(s).

	Hyperparameters
SimCLR	$\tau = 0.07$
MoCo	$K = 65536$,$\tau = 0.07$,$\alpha = 0.99$
BYOL	$\alpha = 0.99$
SimSiam	None
Barlow Twins	$\lambda = 0.005$

Table 4: Hyperparameters for SSL methods in training Causal3dIdent.

	simclr	moco	byol	simsiam	barlow
acc	0.0343	0.0370	0.0190	0.0168	0.0147
0	0.0860	0.0860	0.0392	0.0279	0.0335
1	0.0761	0.0866	0.0476	0.0320	0.0334
2	0.0241	0.0179	0.0178	0.0097	0.0095
3	0.0445	0.0449	0.0164	0.0261	0.0090
4	0.0543	0.0458	0.0300	0.0226	0.0126
5	0.0603	0.0488	0.0262	0.0234	0.0281
6	0.0542	0.0445	0.0420	0.0655	0.0387

Table 5: Accuracy discrepancy between seen and unseen distributions.

We also visualize the latent shift between stable and unstable examples via T-SNE[64]. We employed prediction scores and accuracy to quantitatively demonstrate the effectiveness and value of our proposed solutions, ensuring the soundness of our research and effectively showcasing their benefits.

D. ImageNet

ObjectNet is a large crowdsourced test set for object recognition that includes controls for object rotations, viewpoints, and backgrounds. Objects are posed by workers in their own homes in natural settings according to specific instructions detailing what object class they should use, how and where they should pose the object, and where to image the scene from. Every image is annotated with these properties, allowing us to test how well object detectors work across these conditions. Each of these properties is randomly sampled leading to a much more varied dataset. There are 313 ObjectNet classes with 113 of them overlapping with the ImageNet classes. With each controlled variable, the changes in the variable poses challenges to identify the objects correctly due to the very unusual shift.

Starting from ImageNet **Stylized ImageNet** is constructed by stripping every single image of its original texture and replacing it with the style of a randomly selected painting through AdaIN style transfer. The original objec-

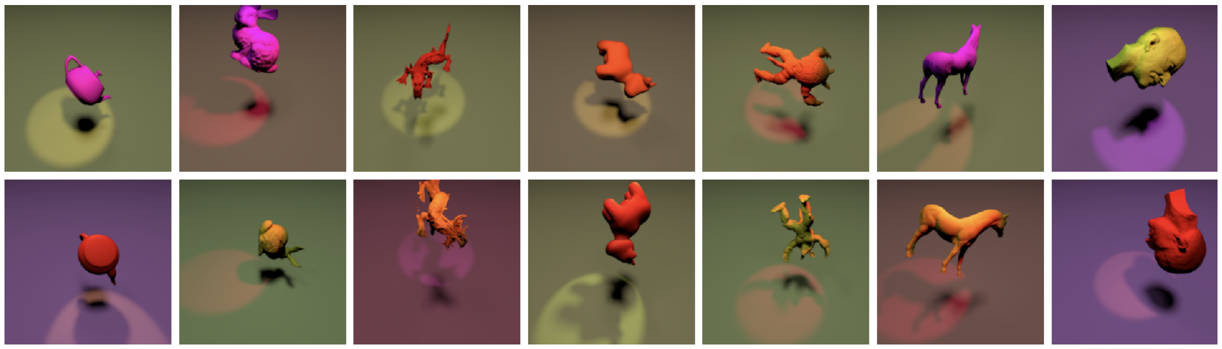
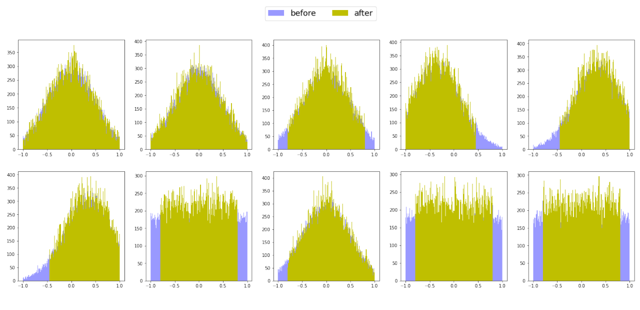
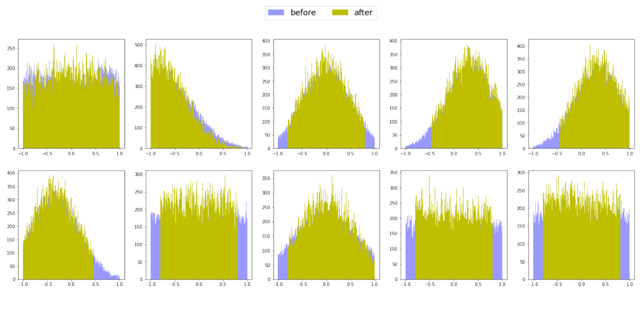


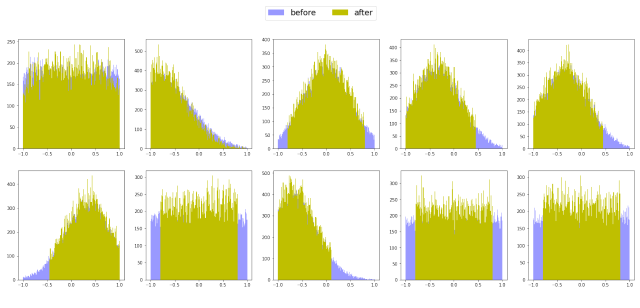
Figure 8: Examples of images in Causal3DIdent. Each image is associated with a 10-dimensional ground-truth latent representation. $[pos_{obj} = (x, y, z), rot_{obj} = (\phi, \theta, \psi), hue_{obj}, pos_{spl}, hue_{spl}, hue_{bg}]$



(a) Teapot



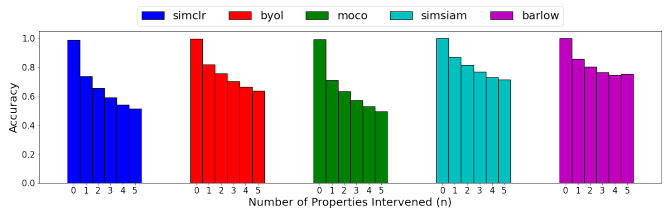
(b) Dragon



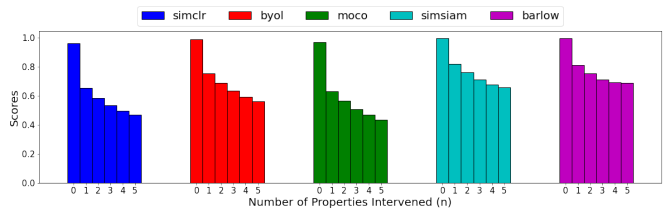
(c) Horse

Figure 9: Examples of sampling intervened data samples. Before sampling and After sampling.

tive for Stylized-ImageNet is to help the network to learn more about shapes, and less about local textures. However,



(a) Deterioration in Accuracy



(b) Deterioration in Score

Figure 10: Deterioration of unstable changing in data variables on all SSL. Only 5 children nodes are intervened.

we regards this shift in the appearance as a change in the data variables(s). Since the stylized images appear drastically different to natural images, we assume this shift is very hard to counter in the representation space due to entangled transformations.

Synthetic Data follows synthetic procedure in [18] where object is masked on a background at a location with a rotation angle. Foreground object masks are cropped from OpenImages [46]. The object classes that overlap with ObjectNet classes are selected, and each class is selected with 10 object masks at highest area and not truncated by any other objects. The backgrounds are sampled from *pexel.com* with the same set of 867 backgrounds used in [18]. Initially there are three data variables to control with: *background*, *rotation*, **location**. The object class is randomly selected at each sythesizing step. At each training step, the other two variables are randomly fixed while

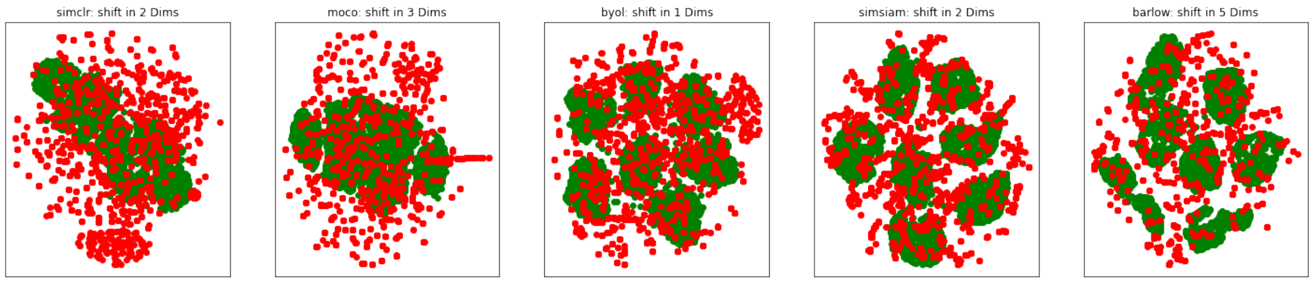


Figure 11: Stable latents are well clustered. Unstable latents are scattered randomly around stable clusters.

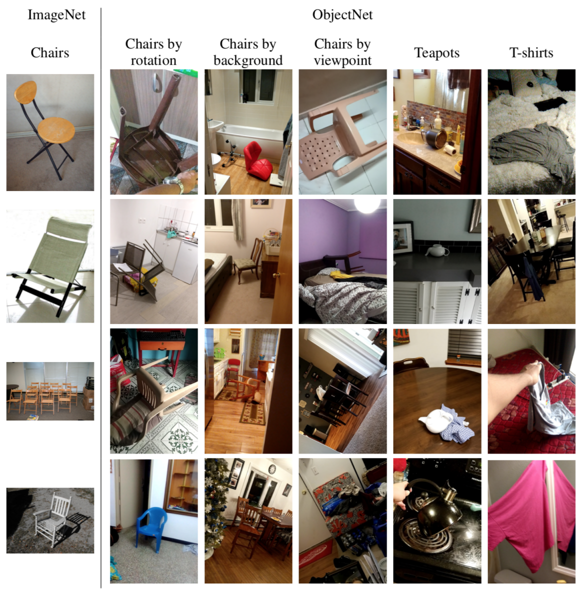


Figure 12: Illustration of ObjectNet controlling data variables.

Figure 13: Examples of Stylized ImageNet

the target variable is randomly selected across 10 values. This results in 10 images to compare with each other. For the special data variable, *texture*, we change the style of the selected object masks to different textures in *Describable Textures Dataset* [13] based on formulation used in **Stylized ImageNet**. With all other three variables randomly fixed, the object mask with 10 different textures are sampled.

We carry the experiment on **Stable Inference Mapping** to 50 epochs and observe the saturation of the improvement (Figure 14). To further improve the performance, more integrated interventions should be applied to make **F** more robust to shifts in the data variables.

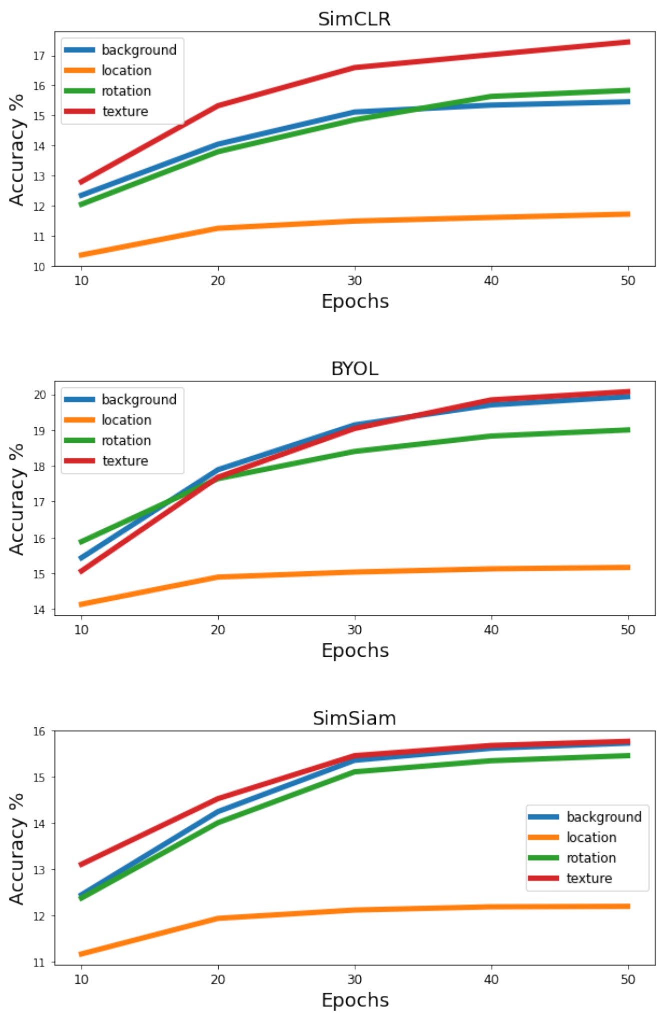


Figure 14: Training longer in **Stable Inference Mapping** can improve the performance. But the improvement saturates after around 30 epochs and the improvement becomes less significant.