We thank all reviewers for their constructive comments and feedback. We have already provided the code in the supplementary material, and will open source it upon acceptance.

To Reviewer 1: **(1) "Optimal z^* is task dependent".** This is not a weakness of our approach, but a point that we emphasize through our analysis in the colored moving MNIST experiments. It indicates that we cannot find a pair of views that are universally optimal for all downstream tasks. The baseline you suggest is included in Table 1 of the supplement, which shows that when all factors (digit, bkdg and pose) are used to create views, the learned z only works well for *background* classification, but does not help *digit classification* and *localization*. This shows that z learned without view selection is not as generalizable as we might think. **(2) "Semi-supervised baselines".** Our focus is on semi-supervised *view* (not feature) learning for verifying the InfoMin hypothesis and supporting our analysis, not to achieve SOTA semi-supervised feature learning. While our contrastive feature learning stage given learned views is unsupervised, we achieved comparable performance as the SOTA semi-supervised learning methods (e.g., on STL-10, our method achieves 5.75% error rate, while MixMatch obtains 5.59%). In the future, our semi-supervised view-learning algorithm could be combined with semi-supervised contrastive representation learning algorithms to further improve performance. **(3) "whether g overfits to a specific task".** The main purpose of learning g with a semi-supervised loss is to verify our InfoMin hypothesis. In theory, it is possible that g makes pre-trained models perform better on tasks similar to the supervised task used to train g, but worse on less similar tasks. We will expand our discussion as suggested. **(4) Figure 2**. Figure 2 is schematic; what are signals and nuisances depends on the downstream task, e.g, signals for one task might be nuisances for another. Empirically we have only observed behavior as depicted Figure 2(a), but in theory Figure 2(b) could also happen. **(5) "No technical contribution on ImageNet augmentation".** Our main goal was to analyze the reverse-U shape phenomenon on a larger-scale and practical data augmentation setup, not to propose new techniques for data augmentation. **(6) "variations between runs for GAN-style training".** Figure 6(a) already includes multiple runs. There is instability in the sense that each single run might end up with a different amount of MI, but the trend of reverse-U shape between MI and accuracy with multiple runs is stable. **(7) "Supervised baseline in Table 2".** Yes, it is trained only on the labeled subset. We will rename the items to make it clearer. **(8) Augmentation in SimCLR** has *not* reached the sweet spot yet. See 'CJ-Blur' (which is SimCLR augmentation) in Figure 4(a) in the supplement.

To Reviewer 2: **(1) "L+ab v.s. image+patch"**. These two setups are not directly comparable since "image+patch" is trained on a different dataset, and please see Sec B.1 in supplement for the reason. Generally, as shown in Table 1, certain views will work better if the shared factors between views are related to the downstream task, as highlighted in our toy MNIST experiments. **(2) "usage of schematic in Figure 1(c)".** One way of making this scheme more practical would be to compute I_{NCE} on smaller models first, or train on a subset of the data to identify good views. This direction deserves further study in future works. **(3) "Correlation of L_{NCE} and downstream accuracy in InstDis".** Thanks for pointing this out, we will note this. We have also clarified in the text that I_{NCE} refers to the *converged* loss, rather than *unconverged* loss along the training. **(4) "how much each augmentation matters".** This is presented in Fig 4 of supplement. We will modify L209 accordingly. **(5) "how g is parameterized".** g consists of several blocks, each with several 1x1 convolutions and relu activations. See B.5 in Supp for more details. **(6) "Figure 2 a schematic?"**. Yes and we will make it clearer in revised version.

To Reviewer 3: **(1) "reverse-U shape corresponds to under-fitting, critical-fitting and over-fitting stages".** This is not true. I_{NCE} in our paper means *converged* loss, not the loss during the training procedure. For each plot, we only vary input views ($\mathbf{v}_1$ and $\mathbf{v}_2$) and train until convergence to get I_{NCE}. We also evaluated I_{NCE} on held-out *validation* data, showing an almost identical reverse-U shape. **(2) "views are defined as linear transformation".** The learned views are more complex: g is a stack of multiple blocks, each consisting of 1x1 convolutions and 'relu' non-linearity (see B.5 and code for details). Note that the view learning experiments is mainly for verifying InfoMin hypothesis. It is still preliminary but is an interesting future direction. **(3) "reasons for not superior to YDbDr composition".** We agree with the explanations R3 provided and leave it as future work. **(4) "Analysis and results on ImageNet".** The core idea of this paper is the InfoMin principle, and its derivative – reverse-U shape. The augmentation analysis on ImageNet is mainly used to support this hypothesis (see Fig 5 in main paper and Fig 4-5 in Supp). We will modify the last contribution accordingly. 'How to learn optimal views (or augmentations)' is an interesting direction but not the primary focus in this paper. We will adopt the suggestion about rewriting Sec. 3.4. **(5) "curves in Figure 4 is not reverse-U shaped?".** This is because there is no other natural color spaces that further reduce MI. So we used learning methods to synthesize neural color spaces with less MI, as shown in Sec 4.2. If you combine the results in Figure 4(a) with Figure 6(a), you will observe the reverse-U shape.

To Reviewer 4: **(1) "trivial solution of $g(\cdot)$".** We avoid such trivial solution by constraining $g(\cdot)$ to be an invertible function, similar to flow-based generative models. Therefore, $g(\cdot)$ is a bijective mapping and total information is preserved after the transformation $g(\cdot)$. **(2) "second view shares a similar location of digit compared to which frame in view 1?"** Given a sequence $\mathbf{x}_{1:20}$, we use the first 10 frames $\mathbf{x}_{1:10}$ as $\mathbf{v}_1$, the digit position of $\mathbf{v}_2$ is the same as the 20-th frame of $\mathbf{x}$, i.e., $\mathbf{x}_{20}$. Therefore, contrastive learning requires the model to extract the position of digits in all 10 frames of $\mathbf{v}_1$ and then extrapolate the motion to predict the digit position in $\mathbf{v}_2$. All position information in $\mathbf{v}_1$ is thus relevant to that in $\mathbf{v}_2$.