Supplementary Materials for “Addressing Feature Suppression in Unsupervised Visual Representations”

Tianhong Li[1,*] Lijie Fan[1,*] Yuan Yuan[1] Hao He[1] Yonglong Tian[1]
Rogerio Feris[2] Piotr Indyk[1] Dina Katabi[1]

[1]MIT CSAIL, [2]MIT-IBM Watson AI Lab

Appendix A: Additional Results

In this section, we provide additional results to better understand PrCL.

Table 1. Comparison of different predictive tasks for PrCL's predictive branch. The table shows the performance of PrCL on Colorful-Moving-MNIST with different predictive tasks in its predictive branch for the fixed feature encoder setting. Colorization achieves good performance on background classification, but bad performance on digit classification since the MNIST digits have no RGB information. Inpainting achieves the best performance among these predictive tasks.

	Colorful-Moving-MNIST	
Recon. Tasks	Digit cls Acc. (%)	Bkgd cls Acc. (%)
No Recon.	15.7	**48.5**
Jigsaw Puzzle	16.1	47.7
Colorization	63.9	47.0
Autoencoder	65.6	42.9
Inpainting	**88.3**	46.5

I. Comparison of Different Predictive Tasks for PrCL's Predictive Branch: In PrCL, we choose the inpainting task for the predictive branch. However, other predictive tasks can be potentially used for the predictive branch. In this section, we evaluate the performance of PrCL with different predictive tasks including Jigsaw Puzzle [6], inpainting, auto-encoder and colorization [9]. Table 1 shows PrCL's performance using different predictive task on Colorful-Moving-MNIST under the fixed feature extractor setting. As shown in the table, all predictive tasks except for Jigsaw Puzzle significantly reduce errors on digit classification in comparison to using contrastive learning without any predictive task. This is because the Jigsaw Puzzle task does not require the features to be able to reconstruct the original image, but just to restore the order of different patches. Learning the background object is sufficient to solve Jigsaw, and the network does not have incentives to learn the digit. The table also shows that inpaintaing compares favorably to other predictive tasks and achieves good performance on both downstream tasks. Hence, we use inpainting as the default predictive task in PrCL.

Table 2. Performance of MoCo on Colorful-Moving-MNIST without masking augmentation (fixed feature encoder setting). The results demonstrate that simply adding masking as a data augmentation does not achieve similar improvements as PrCL.

	Colorful-Moving-MNIST	
Recon. Tasks	Digit cls Acc. (%)	Bkgd cls Acc. (%)
MoCo w/o masking	15.7	**48.5**
MoCo w/ masking	15.2	48.4
PrCL	**88.3**	46.5

II. Masking as a Data Augmentation vs. PrCL: In the predictive branch, PrCL introduces masked input images. Some may wonder whether the improvements are coming from this masking operation, since cutting out the input signals can be viewed as one way of augmentation [2]. However, here in Table 2, we show the performance of MoCo with and without masking augmentation on Colorful-Moving-MNIST (we use the same masking strategy as the predictive branch of PrCL). As shown in the table, the performance of MoCo stays similar with or without masking augmentation. This demonstrate that the improvements of PrCL do not come from this augmentation.

III. Warm-up Training: To show the effectiveness of the proposed warm-up training strategy (Sec. 3 (c)), we compare the results of warm-up training with the results of directly using the combined loss $\mathcal{L}$ (i.e., combining the prediction loss and the contrastive loss) from the beginning. As shown in Table 7, without the warm-up training, on Colorful-Moving-MNIST, PrCL largely degenerates to become similar to the

[*] Indicates equal contribution.

Table 3. Digit classification and background classification accuracy of PrCL with different λ on Colorful-Moving-MNIST dataset.

λ	0	1	5	10	25	50	100	200	500	1000
DIGIT ACC (%)	15.7	48.6	69.8	88.3	88.2	88.3	88.1	87.5	86.3	85.0
BKGD ACC (%)	48.5	47.9	47.5	47.2	47.2	47.1	47.0	45.7	44.5	40.5

Table 4. Performance of PrCL and predictive baselines on MPII for the downstream task of human pose estimation. ↑ indicates the larger the value, the better the performance.

METRIC		Head$^{\uparrow}$	Shoulder$^{\uparrow}$	Elbow$^{\uparrow}$	Wrist$^{\uparrow}$	Hip$^{\uparrow}$	Knee$^{\uparrow}$	Ankle$^{\uparrow}$	PCKh$^{\uparrow}$
FIXED FEATURE EXTRACTOR	Inpainting	83.4	75.2	53.6	44.4	56.4	44.3	45.7	59.0
	Colorization	79.5	71.2	49.6	42.1	54.2	40.7	41.9	55.1
	Autoencoder	79.1	70.1	47.2	41.6	51.9	39.1	40.3	53.8
	PrCL (ours)	**85.7**	**78.8**	**61.7**	**51.3**	**64.4**	**55.6**	**49.2**	**65.1**
	IMPROVEMENTS	**+2.3**	**+3.6**	**+8.1**	**+6.9**	**+8.0**	**+11.3**	**+3.5**	**+6.1**
FINE-TUNING	Inpainting	96.3	**95.2**	87.9	82.1	87.8	82.5	77.6	87.7
	Colorization	96.2	95.1	87.7	82.1	87.8	82.5	77.5	87.6
	Autoencoder	96.0	94.9	87.6	82.0	87.6	82.4	77.3	87.5
	PrCL (ours)	**96.3**	94.9	**88.1**	**82.3**	**87.9**	**82.8**	**77.8**	**87.8**
	IMPROVEMENTS	**+0.0**	**-0.3**	**+0.2**	**+0.2**	**+0.1**	**+0.3**	**+0.2**	**+0.1**

Table 5. Performance on FairFace with PrCL and different predictive unsupervised learning methods. The models are evaluated on downstream tasks of age, gender and ethnicity classification.

METRIC		AGE CLS ACC. (%)	GENDER CLS ACC. (%)	ETHN. CLS ACC. (%)
FIXED FEATURE EXTRACTOR	Inpainting	46.3	83.6	52.9
	Colorization	46.1	82.9	53.8
	Autoencoder	44.3	80.1	50.7
	PrCL (ours)	**50.0**	**87.2**	**61.2**
	IMPROVEMENT	**+3.7**	**+3.6**	**+7.4**
FINE-TUNING	Inpainting	55.0	91.8	68.3
	Colorization	54.9	92.0	68.6
	Autoencoder	54.5	91.3	67.9
	PrCL (ours)	**55.3**	**92.3**	**69.0**
	IMPROVEMENT	**+0.3**	**+0.3**	**+0.4**

Table 6. Performance on Colorful-Moving-MNIST under different methods. The models are evaluated on the downstream tasks of digit classification and background object classification.

METRIC		DIGIT CLS ACC. (%)	BKGD CLS ACC. (%)
FIXED FEATURE EXTRACTOR	Inpainting	84.7	35.0
	Colorization	80.7	38.4
	Autoencoder	81.0	32.9
	PrCL (ours)	**88.3**	**46.5**
	IMPROVEMENT	**+3.2**	**+8.1**
FINE-TUNING	Inpainting	92.9	54.5
	Colorization	92.5	54.5
	Autoencoder	92.4	54.1
	PrCL (ours)	**93.3**	**54.7**
	IMPROVEMENT	**+0.4**	**+0.2**

Table 7. Performance of PrCL on Colorful-Moving-MNIST with and without warm-up training.

	Colorful-Moving-MNIST	
Warm-up Training	DIGIT CLS ACC. (%)	BKGD CLS ACC. (%)
No	24.9	**47.8**
Yes	**88.3**	46.5

contrastive learning baselines and cannot learn good features related to digit classification. This indicates that without the warm-up phase, the contrastive loss can dominate the network causing it to suppress feature at the beginning, and that the network cannot later jump out of the local minimum that suppress feature. On the other hand, with warm-up training, the network first learns a coarse representation; then the contrastive loss helps the network learn more fine-grained representations.

IV. Performance of PrCL with different λ: In PrCL, the combined loss is a weighted average of the prediction loss and the contrastive loss, i.e., $\mathcal{L} = \mathcal{L}_c + \lambda \cdot \mathcal{L}_p$. In the experiments of main paper, λ is set to 10. In this section, we

Table 8. Performance on Colorful-MNIST with progressive removal of data augmentations for different self-supervised learning techniques. The baseline corresponds to the original set of augmentations used in SimCLR and MoCo: random flip, random resized crop, color distortion, and random Gaussian blur.

(a) Experiments on Colorful-MNIST with progressive augmentation removal for Digit classification.

Method	SimCLR		MoCo		BYOL		**PrCL(ours)**		**Improve**
Metric	**Top-1**	**Drop**	**Top-1**	**Drop**	**Top-1**	**Drop**	**Top-1**	**Drop**	
Baseline	14.9	/	15.7	/	15.5	/	**88.3**	/	**+72.6**
Remove flip	14.7	-0.2	15.4	-0.3	15.4	-0.1	**88.1**	-0.2	**+72.7**
Remove blur	14.5	-0.4	15.0	-0.7	15.1	-0.4	**88.1**	-0.2	**+73.0**
Crop color only	13.5	-1.4	14.4	-1.3	13.9	-1.6	**87.9**	-0.4	**+73.5**
Remove color distort	12.4	-2.5	13.1	-2.6	12.8	-2.7	**86.9**	-1.4	**+73.8**
Crop blur only	12.1	-2.8	12.8	-2.9	12.5	-3.0	**86.8**	-1.5	**+74.0**
Crop flip only	12.0	-2.9	12.7	-3.0	12.3	-3.2	**86.7**	-1.6	**+74.0**
Crop only	11.8	-3.1	12.4	-3.3	12.1	-3.4	**86.5**	-1.8	**+74.1**

(b) Experiments on Colorful-MNIST with progressive augmentation removal for Background classification.

Method	SimCLR		MoCO		BYOL		**PrCL(ours)**		**Improve**
Metric	**Top-1**	**Drop**	**Top-1**	**Drop**	**Top-1**	**Drop**	**Top-1**	**Drop**	
Baseline	47.3	/	48.5	/	**49.0**	/	46.5	/	**-2.5**
Remove flip	46.8	-0.5	47.5	-1.0	**47.7**	-1.3	46.4	-0.1	**-1.3**
Remove blur	45.3	-2.0	46.8	-1.7	**47.1**	-1.9	46.3	-0.2	**-0.8**
Crop color only	44.7	-2.6	46.1	-2.4	46.0	-3.0	**46.2**	-0.3	**+0.1**
Remove color distort	43.2	-4.1	45.5	-3.0	45.6	-3.4	**46.0**	-0.5	**+0.4**
Crop blur only	42.3	-5.0	45.3	-3.2	45.2	-3.8	**45.7**	-0.8	**+0.4**
Crop flip only	41.9	-5.4	44.4	-4.1	44.9	-4.1	**45.7**	-0.8	**+0.8**
Crop only	41.5	-5.8	44.1	-4.4	44.4	-4.6	**45.5**	-1.0	**+1.1**

Table 9. Performance comparison on ImageNet classification between PrCL and ConRec[4] with progressive removal of data augmentations.

Method	ConRec[4]	**PrCL(ours)**	**Improve**
Baseline	64.3	**71.0**	**+6.7**
Remove flip	64.1	**70.8**	**+6.7**
Remove blur	63.8	**70.6**	**+6.8**
Crop color only	63.5	**70.1**	**+6.6**
Remove color distort	58.7	**65.9**	**+7.2**
Crop blur only	56.4	**65.1**	**+8.7**
Crop flip only	55.2	**64.6**	**+9.4**
Crop only	54.9	**64.1**	**+9.2**

investigate how different λ affects the performance of PrCL. Note that when $\lambda = 0$, PrCL degenerates to contrastive learning; when $\lambda \to \infty$, PrCL degenerates to predictive learning.

Table 3 compares the performance of PrCL with different λ. As we can see from the results, when $\lambda < 100$, with larger λ, the accuracy of the digit classification increases, while the accuracy of background classification decreases. Moreover, the λ values between 10 and 100 gives quite similar performances, indicating a balancing between contrastive loss and prediction loss. For $\lambda > 100$, the prediction loss dominates the contrastive loss and harm the performance. Therefore, we fix $\lambda = 10$ for all experiments.

V. Predictive Learning vs. PrCL: In the main paper, we mainly compare PrCL with contrastive learning since contrastive learning is the current unsupervised learning SOTA on ImageNet and outperforms predictive learning by a large margin [2, 5]. Here, we also compare PrCL with predictive learning, such as inpainting, colorization and autoencoder, on various datasets to demonstrate the effectiveness of the contrastive branch of PrCL. Tables [4-6] compare PrCL with Inpainting [7], Colorization [9] and Auto-encoder on the RGB datasets. The results demonstrate that PrCL outperforms all predictive learning baselines by a large margin. This is because the contrastive branch in PrCL can significantly improve the quality of the learned representation so it can achieve much better performance on downstream tasks.

VI. Augmentation on Multi-attribute classification In Table 8 we show the performance on multi-attribute classification tasks (on Colorful-MNIST dataset) with progressive removal of data augmentations for different self-supervised learning techniques. From the results we can for traditional contrastive learning methods, the performance for background classification drops a lot as less data augmentations are applied, and the performance for digit classification remains near random no matter what data augmentation techniques are applied. On the contrary the performance for both digit classification and background classification remain almost the same for PrCL when less data augmentation techniques are applied. This is because the additional predictive loss in PrCL could prevent feature suppression.

VII. Comparison with ConRec[4]. In Table 9 we compare our PrCL with ConRec[4] on ImageNet dataset. For ConRec results, we directly reuse the open-source code and keep their training scheme and hyper-parameters. From the results we can see our PrCL outperforms ConRec[4] under every circumstance by a large margin.

Appendix B: Additional Proofs

Here we formally prove Lemma 2.

Lemma 2 (Gradient Equivariance)**.** *The gradient of the empirical infoNCE asymptotics is equivariant under the lifting operation. Formally, consider any lifting operator $\mathcal{T}_\sigma$ from the dimension d_1 to the dimension d_2. We have*

$$\nabla_{\tilde{z_k}} \mathcal{E}_{\texttt{limNCE}}(\mathcal{T}_\sigma(Z); X, t, d_2) = \mathcal{T}_\sigma\left(\nabla_{z_k} \mathcal{E}_{\texttt{limNCE}}(Z; X, t, d_1)\right)$$

Proof.

$$\begin{aligned}
&\nabla_{z_k} \mathcal{E}_{\texttt{limNCE}}(Z; X, t, d_1) \\
&\triangleq \nabla_{z_k}\Big(- \frac{1}{tn^2}\sum_{ij} \lambda_{ij} z_i^\top z_j + \frac{1}{n}\sum_i \log\left(\frac{1}{n}\sum_j e^{z_i^\top z_j/t}\right)\Big) \\
&= -\frac{1}{tn^2}(\sum_{i\neq k} \lambda_{ki} z_i + \sum_{i\neq k} \lambda_{ik} z_i + 2\lambda_{kk} z_k) \\
&+ \frac{1}{tn}\frac{2z_k e^{z_k^\top z_k/t} + \sum_{j\neq k} z_j e^{z_k^\top z_j/t}}{\sum_j e^{z_k^\top z_j/t}} + \frac{1}{tn}\sum_{i\neq k}\frac{z_i e^{z_i^\top z_k/t}}{\sum_j e^{z_i^\top z_j/t}} \\
&= -\frac{1}{tn^2}(\sum_i \lambda_{ki} z_i + \sum_i \lambda_{ik} z_i) \\
&+ \frac{1}{tn}\frac{z_k e^{z_k^\top z_k/t} + \sum_j z_j e^{z_k^\top z_j/t}}{\sum_j e^{z_k^\top z_j/t}} + \frac{1}{tn}\sum_{i\neq k}\frac{z_i e^{z_i^\top z_k/t}}{\sum_j e^{z_i^\top z_j/t}}.
\end{aligned}$$

Since $\mathcal{T}_\sigma$ is a linear operator,

$$\begin{aligned}
&\mathcal{T}_\sigma\left(\nabla_{z_k} \mathcal{E}_{\texttt{limNCE}}(Z; X, t, d_1)\right) \\
&= -\frac{1}{tn^2}(\sum_i \lambda_{ki} \tilde{z_i} + \sum_i \lambda_{ik} \tilde{z_i}) \\
&+ \frac{1}{tn}\frac{\tilde{z_k} e^{z_k^\top z_k/t} + \sum_j \tilde{z_j} e^{z_k^\top z_j/t}}{\sum_j e^{z_k^\top z_j/t}} + \frac{1}{tn}\sum_{i\neq k}\frac{\tilde{z_i} e^{z_i^\top z_k/t}}{\sum_j e^{z_i^\top z_j/t}} \\
&= -\frac{1}{tn^2}(\sum_i \lambda_{ki} \tilde{z_i} + \sum_i \lambda_{ik} \tilde{z_i}) \\
&+ \frac{1}{tn}\frac{\tilde{z_k} e^{\tilde{z_k}^\top \tilde{z_k}/t} + \sum_j \tilde{z_j} e^{\tilde{z_k}^\top \tilde{z_j}/t}}{\sum_j e^{\tilde{z_k}^\top \tilde{z_j}/t}} + \frac{1}{tn}\sum_{i\neq k}\frac{\tilde{z_i} e^{\tilde{z_i}^\top \tilde{z_k}/t}}{\sum_j e^{\tilde{z_i}^\top \tilde{z_j}/t}} \\
&= \nabla_{\tilde{z_k}} \mathcal{E}_{\texttt{limNCE}}(\mathcal{T}_\sigma(Z); X, t, d_2)
\end{aligned}$$

where $\tilde{z_k} = \mathcal{T}_\sigma(z_k)$. The second equality comes from the fact that $\forall z_i, z_j,\ z_i^\top z_j = \mathcal{T}_\sigma(z_i)^\top \mathcal{T}_\sigma(z_j)$. Thus $\nabla_{\tilde{z_k}} \mathcal{E}_{\texttt{limNCE}}(\mathcal{T}_\sigma(Z); X, t, d_2)$ equals $\mathcal{T}_\sigma\left(\nabla_{z_k} \mathcal{E}_{\texttt{limNCE}}(Z; X, t, d_1)\right)$, for all $z_k \in Z$. □

Appendix C: Implementation Details

In this section, we provide the implementation details of the models used in our experiments. All experiments are performed on 8 NVIDIA Titan X Pascal GPUs. On ImageNet, training takes ∼100 hours. On each dataset, we fix the batch size and training epochs for different baselines for a fair comparison. Other parameters for each baseline follow the original paper to optimize for its best performance. Code will also be released upon acceptance of the paper.

ImageNet: We use a standard ResNet-50 for the encoder network. The decoder network is a 11-layer deconvolutional network. The projection head for contrastive learning is a 2-layer non-linear head which embeds the feature into a 128-dimensional unit sphere. The same network structure is used for all baselines and PrCL.

We follow the open repo of [3] for the implementation of MoCo baselines and PrCL on ImageNet. For results on SimCLR, we follow the results reported in [5]. All baselines and PrCL is trained for 800 epochs with a batch size of 256. For the predictive branch of PrCL and the predictive baseline, we mask out 3 to 5 rectangles at random locations in the image. The size of each square is chosen by setting its side randomly between 40 and 80 pixels. For the contrastive branch of PrCL, we apply the same training scheme as MoCo. The first 10 epochs are warm-up epochs, where we only train the network with the prediction loss $\mathcal{L}_p$. For later training, we set $\lambda = 10$. For other RGB datasets, we mainly follows similar implementation as ImageNet.

MPII: We use the network structure similar to the one in [8]. We use a ResNet-50 for the encoder network. Three deconvolutional layers with kernel size 4 and one convolutional layer with kernel size 1 is added on top of the encoded feature to transfer the feature into 13 heatmaps corresponding to 13 keypoints. For the contrastive branch, a 2-layer non-linear projection head is added on top of the encoded feature and embeds the feature into a 128-dimensional unit sphere. For the predictive branch, a decoder network similar to the pose estimation deconvolution network (only the number of output channels is changed to 3) is used to reconstruct the original image. Other implementation details are the same as ImageNet.

For the baselines and PrCL, we train the network for 300 epochs with a batch size of 256. The data augmentation is the same as the baseline augmentations on ImageNet. For PrCL, the first 10 epochs are warm-up epochs, where we only train the network with the prediction loss $\mathcal{L}_p$.

FairFace: We use a standard ResNet-50 for the encoder network. The decoder network is a 11-layer deconvolutional network. The projection head for contrastive learning is a 2-layer non-linear head which embeds the feature into a 128-dimensional unit sphere. The same network structure is used for all baselines and PrCL.

For the baselines and PrCL, we train the network for 1000 epochs with a batch size of 256. The data augmentation is the same as the baseline augmentations on ImageNet. For PrCL, the first 30 epochs are warm-up epochs, where we only train the network with the prediction loss $\mathcal{L}_p$.

Colorful-Moving-MNIST: We use a 6-layer ConvNet for the encoder. The encoder weights for the predictive and contrastive branches are shared. The decoder is a 6-layer deconvolutional network symmetric to the encoder. The projection head for contrastive learning is a 2-layer non-linear head which embeds the feature into a 64-dim normalized space.

We use the SGD optimizer with 0.1 learning rate, 1e-4 weight decay, and 0.9 momentum to train the model for 200 epochs. The learning rate is scaled with a factor of 0.1 at epoch 150 and 175. The batch size is set to 512. The temperature for contrastive loss is set to 0.1. For PrCL, the first 30 epochs are warm-up epochs, where we only train the network with the prediction loss $\mathcal{L}_p$.

For PrCL, for each input image after augmentation with a size of 64 by 64 pixels, we randomly mask out 3 to 5 rectangle patches at random locations in the image and fill them with the average pixel value of the dataset. The size of each square is chosen by setting its side randomly between 10 and 16 pixels.

Appendix D: Evaluation Metrics

For ImageNet, FairFace and Colorful-Moving-MNIST, the evaluation metrics are the standard Top-1 classification accuracy. For MPII, we evaluate the learned representations under the single pose estimation setting [1]. Each person is cropped using the approximate location and scale provided by the dataset. Similar to prior works, we report the PCKh (Percentage of Correct Keypoints that uses the matching threshold as 50% of the head segment length) value of each keypoint and an overall weighted averaged PCKh over all keypoints (head, shoulder, elbow, wrist, hip, knee, ankle).

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[4] Jonas Dippel, Steffen Vogler, and Johannes Höhne. Towards fine-grained visual representations by combining contrastive learning with image reconstruction and attention-weighted pooling. *arXiv preprint arXiv:2104.04323*, 2021.

[5] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[6] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[7] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[8] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[9] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.