A Appendix

A.1 Proof of 3.3

Proof. To optimize the objective in 8 with the standard widely available optimizers, we adopt the following efficient mini-batch stochastic gradient estimator.

$$\begin{aligned}
\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i) &= 2\left[(\mathrm{sim}(i,i+) - a)\nabla_{\mathbf{w}}\mathrm{sim}(i,i+) \mid y_{ii+}\right] \\
&\quad + 2\sum_{j\in\mathcal{B}_i} \left[2(\mathrm{sim}(i,j)-b)\nabla_{\mathbf{w}}\mathrm{sim}(i,j) \mid y_{ij-}\right] + 2\alpha\big[-\nabla_{\mathbf{w}}\mathrm{sim}(i,i+)\mid y_{ii+}\big] \\
&\quad + 2\alpha\Big[\sum_{j\in\mathcal{B}_i}\nabla_{\mathbf{w}}\mathrm{sim}(i,j)\mid y_{ij-}\Big], \qquad (9) \\
\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i) &= 2\sum_{j\in\mathcal{B}_i}\big[(b-\mathrm{sim}(i,j))\mid y_{ij-}\big],
\end{aligned}$$

where $\mathcal{B}_i$ is a random mini-batch of images drawn from $\mathcal{M}_i$, which still excludes the image $\mathbf{x}_i$ itself. For the analysis, we consider the following simple update

$$\begin{aligned}
\mathbf{w}_{t+1} =& \mathbf{w}_t - \frac{\beta}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i) \\
b_{t+1} =& b_t - \frac{\beta}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i)\ . \qquad (10)
\end{aligned}$$

Our analysis adopts the following assumption. Note that the cosine similarity metric $\mathrm{sim}(i,j)$ for any two augmentations x_i, x_j and two operations $\mathcal{A}, \mathcal{A}'$ is a function of $\mathbf{w}$. Then, for the purpose of analysis, we instead use the notation $\mathrm{sim}(i,j)(\mathbf{w})$ to capture the dependence on $\mathbf{w}$.

Assumption A.1. We assume for any $i, j, \mathcal{A}, \mathcal{A}'$, the cosine similarity metric $\mathrm{sim}(i,j)(\mathbf{w})$ satisfies

- $\mathrm{sim}(i,j)(\mathbf{w})$ is L_0-Lipschitz continuous and L_1-smooth.
- $\max\{a, b\} \le \tau$,

for some positive constant $\tau > 0$.

Note that the boundedness condition in the second item can be replaced by adding a projection of $\mathbf{w}$ and b onto a bounded set like a ball for the updates in 10. However, for the simplicity, we directly assume the boundedness, which is also observed during the optimization process in the experiments. The global contrastive objective in our case that is based on the entire dataset $\mathcal{D}$ is is given by $\min_{\mathbf{w}\in\mathbf{R}^{\mathbf{d}}, b}\mathbb{E}_{\mathbf{x}_i\sim\mathcal{D},\mathcal{A},\mathcal{A}'\sim\mathcal{P}}[L_s'(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i)]$ where

$$\begin{aligned}
L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) = \big[(\mathrm{sim}(i,i+) - a)^2 \mid y_{ii+}\big] &+ \sum_{j\in\mathcal{M}_i}\big[\mathrm{sim}(i,j) - b)^2 \mid y_{ij-}\big] \\
&+ \Big\{2\alpha\big[1 - \mathrm{sim}(i,i+) \mid y_{ii+} + \sum_{j\in\mathcal{M}_i}\mathrm{sim}(i,j)\mid y_{ij-}\big] - \alpha^2\Big\}.
\end{aligned}$$

We first prove the smoothness of this objective function. Note that the cosine similarity $\mathrm{sim}(i,j)(\mathbf{w})$ is bounded by 1, and hence we have $\max\{|\mathrm{sim}(i,j)(\mathbf{w})|, a, b\} \le \tau + 1$. Then, the gradient of the this objective takes the form of

$$\begin{aligned}
\nabla_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) &= \left[2(\mathrm{sim}(i,i+) - a)\nabla_{\mathbf{w}}\mathrm{sim}(i,i+) \mid y_{ii+}\right] \\
&\quad + \sum_{j\in\mathcal{M}_i}\left[2(\mathrm{sim}(i,j)-b)\nabla_{\mathbf{w}}\mathrm{sim}(i,j) \mid y_{ij-}\right] \\
&\quad + 2\alpha\big[-\nabla_{\mathbf{w}}\mathrm{sim}(i,i+)\mid y_{ii+}\big] + 2\alpha\Big[\sum_{j\in\mathcal{M}_i}\nabla_{\mathbf{w}}\mathrm{sim}(i,j)\mid y_{ij-}\Big], \\
\nabla_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) &= 2\sum_{j\in\mathcal{M}_i}\big[(b-\mathrm{sim}(i,j))\mid y_{ij-}\big].
\end{aligned}$$

Based on the gradient form here, we can obtain, for any two parameters $(\mathbf{w}, b)$, $(\mathbf{w}', b')$

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) - \nabla_{\mathbf{w}} L(\mathbf{w}', b'; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i)\| \\
&\quad \leq \Big\| \, [2(\text{sim}(i, i{+})(\mathbf{w}) - a)\nabla_{\mathbf{w}} \text{sim}(i, i{+})(\mathbf{w}) \mid y_{ii+}] - \\
&\qquad [2(\text{sim}(i, i{+})(\mathbf{w}') - a)\nabla_{\mathbf{w}} \text{sim}(i, i{+})(\mathbf{w}') \mid y_{ii+}] \, \Big\| \\
&\qquad + \sum_{j \in \mathcal{M}_i} \| \, [2(\text{sim}(i, j)(\mathbf{w}) - b)\nabla_{\mathbf{w}} \text{sim}(i, j)(\mathbf{w}) \mid y_{ij-}] - \\
&\qquad [2(\text{sim}(i, j)(\mathbf{w}') - b')\nabla_{\mathbf{w}} \text{sim}(i, j)(\mathbf{w}') \mid y_{ij-}] \, \| \\
&\qquad + 2\alpha \Big\| \big[- \nabla_{\mathbf{w}} \text{sim}(i, i{+})(\mathbf{w}) \mid y_{ii+} \big] - \big[- \nabla_{\mathbf{w}} \text{sim}(i, i{+})(\mathbf{w}') \mid y_{ii+} \big] \Big\| \\
&\qquad + 2\alpha \sum_{j \in \mathcal{M}_i} \| \big[\nabla_{\mathbf{w}} \text{sim}(i, j)(\mathbf{w}) \mid y_{ij-} \big] - \big[\nabla_{\mathbf{w}} \text{sim}(i, j)(\mathbf{w}') \mid y_{ij-} \big] \|,
\end{aligned}$$

which, in conjunction with A.1 and using the fact that $ab - a'b' = a(b - b') + (a - a')b'$, yields

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) - \nabla_{\mathbf{w}} L(\mathbf{w}', b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'), \mathcal{M}_i\| \\
&\quad \leq 4(\tau + 1)L_1 n \|\mathbf{w} - \mathbf{w}'\| + 2L_0^2 n \|\mathbf{w} - \mathbf{w}'\| + 2(n - 1)L_0 |b - b'| + 2\alpha n L_1 \|\mathbf{w} - \mathbf{w}'\| \\
&\quad \leq (4(\tau + 1)L_1 n + 2L_0^2 n + 2\alpha n L_1)\|\mathbf{w} - \mathbf{w}'\| + 2(n - 1)L_0 |b - b'| \\
&\quad \leq \underbrace{\sqrt{2}(4(\tau + 1)L_1 n + 2L_0(L_0 + 1)n + 2\alpha n L_1)}_{L_w} \sqrt{\|\mathbf{w} - \mathbf{w}'\|^2 + |b - b'|^2}
\end{aligned} \tag{11}$$

Similarly, for the gradient w.r.t. b, we have

$$\begin{aligned}
\|\nabla_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i) - \nabla_b L(\mathbf{w}', b'; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{M}_i)\| & \\
\leq 2(n - 1)|b - b'| + 2(n - 1)L_0 \|\mathbf{w} - \mathbf{w}'\| & \\
\leq \underbrace{2n(L_0 + 1)\sqrt{2}}_{L_b} \sqrt{\|\mathbf{w} - \mathbf{w}'\|^2 + |b - b'|^2} &
\end{aligned} \tag{12}$$

First note that our stochastic gradient estimator $\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i)$ and $\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i)$ are unbiased estimators. To see this, based on the forms in 9, we have

$$\begin{aligned}
\mathbb{E}\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i) &= \mathbb{E}[\mathbb{E}\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i) \mid x_i, \mathcal{A}, \mathcal{A}'] \\
&= \mathbb{E}[\mathbb{E}\nabla_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}') \mid x_i, \mathcal{A}, \mathcal{A}'] \\
&= \nabla_{\mathbf{w}} L(\mathbf{w}, b)
\end{aligned}$$

where the first equality follows because $\mathcal{B}_i$ is sampled from $\mathcal{M}_i$. A similar result is obtained for $\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i)$, i.e., $\mathbb{E}\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i) = \nabla_b L(\mathbf{w}, b)$.

Based on the smoothness results in 11 and 12 and the unbiased estimation, we are now ready to prove the main theorem. Let $\mathbf{v} = (\mathbf{w}, b)$ denote all optimization parameters. From the smoothness results in 11 and 12, we can establish the smoothness of the overall objective $L(\mathbf{v}) = L(\mathbf{w}, b)$ as below. For any $\mathbf{v}$ and $\mathbf{v}'$,

$$\|\nabla L(\mathbf{v}) - \nabla L(\mathbf{v}')\| \leq \sqrt{L_b^2 + L_w^2} \|\mathbf{v} - \mathbf{v}'\|. \tag{13}$$

Then, based on 13, we have

$$L(\mathbf{v}_{t+1}) \leq L(\mathbf{v}_t) + \langle \mathbf{v}_{t+1} - \mathbf{v}_t, \nabla L(\mathbf{v}_t) \rangle + \frac{\sqrt{L_b^2 + L_w^2}}{2} \|\mathbf{v}_{t+1} - \mathbf{v}_t\|^2,$$

which, by taking the expectation $\mathbb{E}_t := \mathbb{E}[\cdot|v_t]$ on the both sides and using our unbiased gradient estimators, yields

$$\begin{aligned}\mathbb{E}_t L(\mathbf{v}_{t+1}) \leq & L(\mathbf{v}_t) - \beta\|\nabla L(\mathbf{v}_t)\|^2 + \frac{\sqrt{L_b^2 + L_w^2}}{2}\mathbb{E}_t\|\mathbf{v}_{t+1} - \mathbf{v}_t\|^2 \\ = & L(\mathbf{v}_t) - \beta\|\nabla L(\mathbf{v}_t)\|^2 + \frac{\sqrt{L_b^2 + L_w^2}}{2}\beta^2\mathbb{E}_t\Big(\Big\|\frac{1}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}', \mathcal{B}_i)\Big\|^2 \\ & + \Big\|\frac{1}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i)\Big\|^2\Big). \end{aligned} \tag{14}$$

Note from the forms in 9 and A.1, we have

$$\begin{aligned}\Big\|\frac{1}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i)\Big\|^2 \leq & \Big\|\widehat{\nabla}_{\mathbf{w}} L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i)\Big\|^2 \\ \leq & 8(\tau+1)nL_0 + 2\alpha nL_0 \\ \Big\|\frac{1}{B}\sum_{i\in\mathcal{B}}\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i)\Big\|^2 \leq & \Big\|\widehat{\nabla}_b L(\mathbf{w}, b; \mathbf{x}_i, \mathcal{A}, \mathcal{A}'; \mathcal{B}_i)\Big\|^2 \leq 4n(\tau+1). \end{aligned} \tag{15}$$

Incorporating 15 into 14 yields

$$\mathbb{E}_t L(\mathbf{v}_{t+1}) \leq L(\mathbf{v}_t) - \beta\|\nabla L(\mathbf{v}_t)\|^2 + \beta^2\frac{\sqrt{L_b^2 + L_w^2}}{2}(4n(\tau+1) + 8(\tau+1)nL_0 + 2\alpha nL_0).$$

Unconditioning on $\mathbf{v}_t$, rearranging the above inequality and doing the telescoping over t from 0 to $T-1$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla L(\mathbf{v}_t)\|^2 \leq & \frac{L(\mathbf{v}_0) - \min_{\mathbf{v}} L(\mathbf{v})}{\beta T} + \beta\frac{\sqrt{L_b^2 + L_w^2}}{2}(4n(\tau+1) + 8(\tau+1)nL_0 + 2\alpha nL_0) \\ \leq & \mathcal{O}(\frac{1}{\beta T} + \beta) \end{aligned} \tag{16}$$

which, in conjunction with $\beta = \frac{1}{\sqrt{T}}$ and the definition of t', finishes the proof. □

A.2 AUC OPTIMIZATION AND CONTRASTIVE LEARNING

We further elucidate our motivations behind adaptation of the AUC optimization framework towards contrastive learning. AUC as a metric was formulated for binary classification wherein, the objective of the network is to enhance the prediction scores for “positive” samples in comparison to the “negatives” (Equation 3.1). Thus by virtue of its construction, it aligns seamlessly for an application in contrastive learning wherein due to the lack of labels, one is compelled to enforce separation amongst classes through a binary objective with "positives" being the augmentations of the same sample and "negatives", the augmentations of other samples within the batch. Additionally, AUC was originally devised to address the imbalance of classes whereby accuracy as a metric may lead to misleading evaluation of the network. A classic example of this phenomenon is often cited with a dataset containing 100 samples, 99 of which are of the “positive” class and a network that predicts every sample as a “positive” will therefore have attained a 99% accuracy. This aspect of the function resonates well with the context of contrastive learning in our application, as for one image in our batch of samples, the remaining images are considered to be “negative”.

A.3 ADDITIONAL RESULTS

A.3.1 LONGER PRETRAINING

In order to attain stronger convergence, we conduct the pretraining procedure using our method for 800 epochs on ImageNet with the ResNet-50 backbone. In Table 6, we compare against the prominent methods for SSL. We retain the batch size of 256 for our method, wherein the remaining methods have been trained using a larger batch size of 1024. The parameters and setup of the method follows the description in 4. Yet again, our loss function avails a superior result using a far smaller batch size.

Table 6: **ImageNet Longer Pretraining (ResNet-50)**. Top-1 accuracy for linear evaluation results are listed. We conduct pretraining for 800 epochs and compare against known arts.

Method	Batch Size	800 ep
SimCLR Chen et al. (2020a)	1024	69.1
Moco-v2 Chen et al. (2020b)	1024	71.1
InfoMin Poole et al. (2020)	1024	73.0
BarlowTwins Zbontar et al. (2021)	1024	73.2
OBOW Gidaris et al. (2020)	1024	73.8
BYOL Grill et al. (2020)	1024	74.4
DCv2 Caron et al. (2018)	1024	75.2
SwAV Caron et al. (2020)	1024	75.3
DINO Caron et al. (2021)	1024	75.3
Ours (400 ep)	256	73.5
Ours (800 ep)	256	**75.5**

A.3.2 Comparison against MoCo-v3

Pretraining We train the MoCo-v3 architecture by replacing the objective function with ours whilst retaining the architecture and parameter settings. Our results are based on a batch size of 128 on the datasets Cifar-10, ImageNet-S and ImageNet using the ViT-small backbone, which are illustrated in Table 7. Our models were trained for 100 epochs with the default parameter and augmentation settings. Our model consistently outperforms the results when comparing to MoCo-v3 by values higher than 2% for Cifar-10 and Cifar-100 datasets and by over 5% for ImageNet.

Table 7: **Pre-training and linear evaluation vs MoCo-v3.** 'a/b/c' in the kNN column are acc. (%) with k = 10, 20, 100, respectively. The Cifar results use the ResNet-18 backbone and the ImageNet results use the Vit-Small backbone.

	MoCo-v3		Ours	
Dataset	KNN	Linear	KNN	Linear
Cifar-10	84.6/84.5/84.4	91.4	87.6/87.2/86.1	93.6
Cifar-100	51.3/53.5/52.7	66.6	52.9/54.7/53.4	69.7
ImageNet-100	74.1/74.7/73.8	77.6	77.0/77.6/76.9	82.5
ImageNet	50.0/50.6/49.1	62.3	54.1/54.0/51.4	67.9

Transfer Learning We subsequently evaluate the model for transfer learning on the Cifar-10, Cifar-100 datasets, Flowers-102 Nilsback & Zisserman (2008) and the Pets Parkhi et al. (2012) datasets after pretraining on ImageNet. The protocol followed is identical as mentioned in Chen et al. (2021) and the results are listed under Table 8

Table 8: **Transfer Learning comparison with MoCov3** after pretraining on ImageNet. Numbers next to the method indicate the batch size used. 'a/b' represent the Top-1 and Top-5 accuracies (%). The numbers listed under the 'Supervised' category are borrowed from Dosovitskiy et al. (2020).

	Cifar-10	Cifar-100	Flowers-102	Pets	Average
Random Init.	77.8	48.5	54.4	40.1	55.2
Supervised	98.1	87.1	89.5	93.8	92.1
Moco-v3-256	97.1/100.0	84.6/97.7	88.6/96.7	85.0/98.6	88.8/98.2
Ours-128	98.2/100.0	85.4/97.9	88.9/97.5	87.1/99.3	89.9/98.7

A.3.3 Comparison against SimCLR

We illustrate the performance of our method on Cifar-10, Cifar-100 and STL-10 in Table 9, comparing against SimCLR for varying batch sizes and epochs. In this experiment, we use a backbone of ResNet-50. As is standard practice, we replace the first 7 × 7 Conv layer with stride 2 with 3 × 3 with stride 1

Table 9: **SimCLR comparison.** Top-1 KNN evaluation results on Cifar-10, Cifar-100 and STL-10 datasets.

	BS	64			128			256		
Dataset	Epoch	200	300	500	200	300	500	200	300	500
Cifar-10	SimCLR	79.6	81.8	84.15	82.8	83.4	86.2	83.8	85.8	86.9
	Ours	85.4	87.4	89.1	86.1	87.9	89.4	85.7	87.3	88.7
Cifar-100	SimCLR	45.2	47.8	49.9	47.8	53.5	55.6	48.5	54.1	56.0
	Ours	52.9	56.0	57.2	53.7	56.1	57.3	53.7	55.9	57.1
STL-10	SimCLR	69.8	69.7	73.6	72.1	72.9	74.1	75.6	75.7	76.2
	Ours	76.1	76.2	78.6	77.5	78.9	80.9	76.8	78.4	80.4

Table 10: **Comparison against DCL** for the listed batch sizes and datasets. The linear evaluation scores following the protocol mentioned in DCL are listed. All models are trained for 200 epochs. The backbone used for ImageNet was ResNet-50 whereas for the other datasets, the backbone was set to ResNet-18.

Method	BS/Dataset	128	256	512
DCL	ImageNet	64.3	65.9	65.8
Ours	ImageNet	67.6	67.7	67.9
DCL	Cifar-10	85.7	85.3	84.7
Ours	Cifar-10	88.4	87.9	88.3
DCL	Cifar-100	58.9	58.5	58.4
Ours	Cifar-100	59.3	59.5	59.5
DCL	STL-10	86.1	85.7	85.6
Ours	STL-10	86.5	86.3	86.5

and remove the first max-pooling layer. The learning rate is fixed to $1e-3$. The training is conducted for 500 epochs. We notice that our method significantly outperforms SimCLR for smaller batch sizes by margins of $+7\%$ for both datasets, and retains a consistent performance across the batch sizes. Upon convergence, our method outperforms SimCLR in Top-1 accuracy by an average of 3.8%. Subsequent to the epoch 20, our method overcomes the performance of SimCLR, and towards the end of training outperforms by over 3.2%. This trend is reflected across the datasets and batch-size, epoch settings, with the margins of out-performance particularly stark for smaller batch sizes.

A.3.4 Comparison against DCL

We compare against the popular objective of DCL Yeh et al. (2022) directly for the datasets of ImageNet, Cifar and STL. Here, identical to their work, we pretrain ResNet-50 and ResNet-18 architectures for ImageNet and the rest respectively, for 200 epochs. We follow the augmentation parameters as per their work in order to retain fairness and train our models for varying batch sizes. The results are listed in Table 10. We again witness a substantial margin of improvement over DCL across the comparisons.

A.3.5 Robustness to α

We conduct extensive ablations to experiment with the significance of the component A_3 in our main formulation 8. We train the model using the parameters and architecture described in section A.3.3 with a batch size of 128 on Cifar-10, while varying the value of the parameter α which modulates the influence of the component. The results are illustrated in Table 11. Here we show the KNN evaluation results for at several epochs during training retaining identical settings. We establish that A_3 is crucial to our formulation and prevents the mode collapse phenomenon often observed in SSL frameworks, where the features are mapped to a unique point in the hypersphere regardless of the

Table 11: **Robustness to α.** We evaluate our loss function on the Cifar-10 dataset for varying values of α using the KNN protocol for k=200. The values are the Top-1 KNN evaluation accuracies at various epochs and α parameter settings.

Epoch/α	0.0	0.1	0.5	0.7	1.0
50	25.1	77.7	78.3	77.6	76.8
100	10.0	82.9	83.4	82.4	81.5
150	10.00	85.18	85.70	84.76	84.41
200	10.00	86.80	86.61	85.89	85.98
250	10.00	87.40	87.64	87.21	87.06

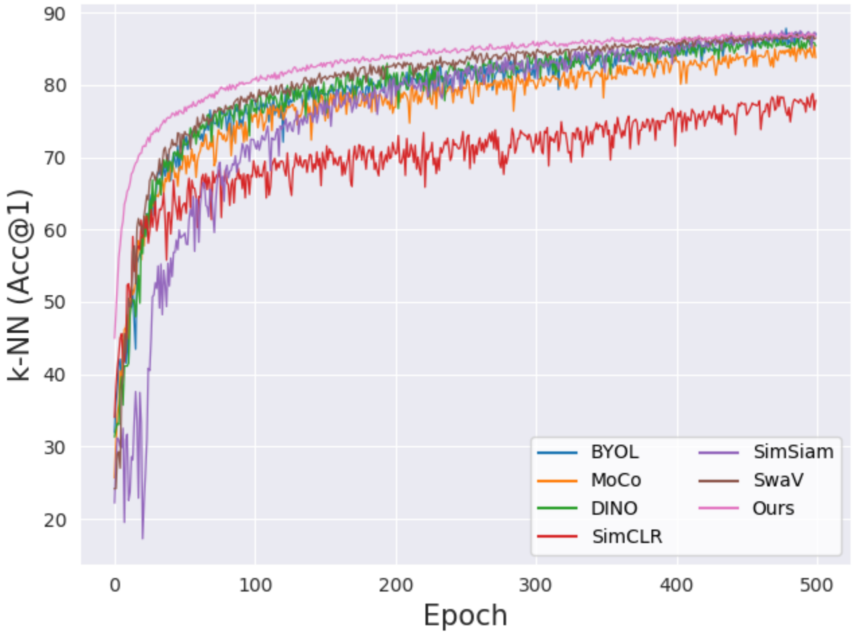

Figure 3: ***k*-NN curves:** Plot of the k-NN accuracy curves for various methods trained on Cifar-10 for 500 epochs using ResNet-18. The values for the other methods were borrowed from LightlyAI

class distinction. Moreover, for all other values of α, our formulation retains its performance across the epochs, with nominal differences, thus illustrating that the component is crucial to the formulation as well as robust to variations in α, which therefore requires no additional tuning.