We thank the four reviewers for their constructive comments. The following are our responses to reviewers' comments. (We use T, SHT, NNP, and MS to denote triplet, semihard triplet, normalized n-pair, and multi-similarity, respectively.)

To Reviewer #2 Q1: Some formulations are confusing. **R1**: Thanks. We will rewrite the formulations in the revision.

Q2: More applications beyond metric learning? More discussions? **R2**: We add experiments of SimCLR on CIFAR10 in Table 1. The classifier is trained with lr in $\{2, 5, 10\}$ and bs=256 for 50 epochs. We will also discuss more related works accordingly, e.g., contrastive learning in unsupervised/self-supervised learning as suggested by the reviewer.

To Reviewer #3 Q1: Difference with previous regularization methods? One more ablation study? **R1**: (1) Motivation: The original N-pair loss uses inner product without l_2-normalization as the similarity measure, which aims to optimize only the direction and remove the influence of norms. However, we consider losses with l_2-normalization to alleviate the unbalanced direction update caused by large norm variance. (2) Formulation: The l_2-regularizer in N-pair loss constrains the norms to be small, while SEC reduces the norm variance and is more effective as shown in Table 2. Besides, for the method Clustering, it only uses the common l_2-normalization.

Q2: Empirical results of unstable update gradient? **R2**: Since it's difficult to directly show the gradient, in Figure 1 we provide the unstable change of samples' norms, which are the denominator factors of gradient magnitudes of corresponding embeddings, to reflect this instability.

Q3: Hyper-parameters and how to determine them? code? **R3**: The training settings are in Table 3. The hyper-parameters in losses follow [1] (Section 5) for T, the original paper for SHT, [2] for NNP (we test s=25 and 64), and original authors' GitHub for MS. Other settings such as network structure are following the paper of MS. We will release the code once this paper is accepted.

Q4: A brief description of each prior work. **R4**: We will rewrite more detailly in the revision.

Q5: Normalized n-pair loss? **R5**: Thanks. It actually stands for Equation 3 in our paper. We add experiments of tuplet margin loss (TML) (w.o. and with SEC) in Table 4. We use hyper-parameters ($\beta = 0.1, \lambda = 0.5, \epsilon = 0.01$) in the original paper, bs=128 (4 instances/class), and Adam.

Q6: More broader impact discussions. **R6**: We will add more discussions in the revision.

To Reviewer #4 Q1: The straightforward and trivial design of SEC. **R1**: Thanks. Though the formulation is straightforward, the underlying goal of SEC is not trivial, aiming to adjust the gradient contributions from different embeddings. In particular, we introduce a novel perspective of the impact of large norm variance for angular loss optimization, which offers an important guidance for the related algorithm design both theoretically and empirically. Further, compared to another l_2-regularizer, the experiments show that SEC is a better choice (please see Reviewer #3' R1 for details) and is useful for many different kinds of angular losses.

Q2: The calculation of average norm. **R2**: Thanks. In practice, we only calculate the average norm in a mini-batch. From Figure 2, we observe that the averaged norm is smoothly changing and finally stable.

Q3: Compare with an intuitive baseline? **R3**: The baseline losses in Table 3 and 4 in our paper have already operated on the l_2-normalized features and SEC is designed to reduce the norm variance of embeddings when using l_2-normalization.

Table 1: SimCLR with SEC (ResNet50 encoder, linear head, NT-Xent, dim=128, bs=256, temp=0.5, SGD. Best accuracy of the classifier is reported).

Method	Epoch	LR	Top 1
SimCLR (w.o. SEC)	150	cosine decay (0.1-0.5 grid search)	86.63
SimCLR (with 0.1*SEC)	150	cosine decay (0.1-0.5 grid search)	**87.00**

Table 2: Comparisons of l_2-norm regularizers (Cars, SHT).

η	μ	NMI	F1	R@1	Mean	Var.
0	-	67.64	38.31	80.17	7.63	2.82
0.5	avg	**72.67**	**44.67**	**85.19**	3.20	0.03
0.1	0	64.57	34.72	75.78	0.45	0.01
0.01	0	68.05	38.49	80.61	1.41	0.07
0.005	0	69.24	40.24	82.60	1.98	0.14
0.001	0	69.13	40.62	81.54	4.36	0.79

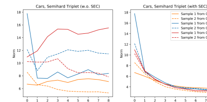

Figure 1: The changing curves of several samples' norms.

Table 3: Hyper-parameters.

Dataset	Iters	Loss	[lr for backbone]/[lr for head]/[lr decay@iter]
CUB	8k	T, SHT	5e-6/2.5e-6/0.1@5k
		NNP	1e-5/5e-6/0.1@5k
		MS	5e-5/2.5e-5/0.1@5k
Cars	8k	T, SHT, NNP	1e-5/1e-5/0.5@4k, 6k
		MS	4e-5/4e-5/0.1@2k
SOP, In-shop	12k	T, SHT, NNP, MS	5e-4/1e-4/0.1@6k

Table 4: TML with SEC.

Dataset	LR	Method	NMI	F1	R@1
CUB	4e-5/2e-5/ 0.1@5k	TML	68.96	39.25	63.64
		+SEC	**71.00**	**42.28**	**64.74**
Cars	6e-5/6e-5/ 0.5@4k, 6k	TML	69.78	41.12	82.87
		+SEC	**72.77**	**43.03**	**84.20**
SOP	5e-4/1e-4/ 0.1@6k	TML	**90.50**	**38.45**	73.66
		+SEC	90.45	37.92	**74.34**
			R@1	R@2	R@3
In-shop	1e-3/2e-4/ 0.1@6k	TML	84.50	96.95	98.09
		+SEC	**84.74**	**97.38**	**98.27**

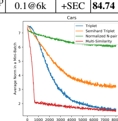

Figure 2: The average norm in a mini-batch at each iteration.

Table 5: SEC with BN/WN.

Method	NMI	F1	R@1	Mean	Var.
SEC	**72.67**	**44.67**	**85.19**	3.20	0.03
BN	67.09	37.31	80.20	7.77	3.00
BN+SEC	47.71	15.71	62.46	7.01	0.03
WN	66.98	37.47	79.42	9.31	4.95
WN+SEC	71.81	43.10	85.01	3.18	0.03

Q4: Larger improvements come from larger variance reduction? **R4**: The variance reduction on Cars training set: 5.77$\rightarrow$0.02 for T, 2.82$\rightarrow$0.03 for SHT, 1.76$\rightarrow$0.13 for NNP, and 1.68$\rightarrow$0.004 for MS, thus this conclusion makes sense.

Q5: More broader impact discussions. **R5**: We will add more discussions in the revision.

To Reviewer #5 Q1: Part of the analysis/theory are known. **R1**: Thank you for the comment, however, for Proposition 1, we believe that the Section 3 and Figure 3 in [3] haven't show that $\frac{\partial L}{\partial f}$ is vertical to f. For Proposition 2, we agree that the Section 3.3 in [4] also mentions that the magnitude of the gradient is inversely proportional to the embedding norm (we will add it to related works), however, we take a further step by explaining how this gradient influences the direction update and how to solve the problem, which are ignored by [4].

Q2: More analysis about the optimizer? **R2**: Thanks. Adam (and other optimizers with adaptive lr) will adjust the lr for each model parameter according to its historical gradient magnitude, resulting in current gradient magnitude changed. We think it helps balance the update of each parameter in some extent. However, for each individual parameter, Adam would not further analyze its gradient compositions from different embeddings and separately adjust these components considering the influence of different embedding norms. Therefore, we suspect that Adam would alleviate this problem to some extent, but the unbalanced direction update among embeddings caused by large norm variance still exists.

Q3: Interplay of SEC and batch norm/weight norm? **R3**: Figure 1 is generated without BN on top of the final embedding and we add two contrast experiments: (1) adding BN on top of the final embedding before l_2-normalization (2) employing weight normalization for the final fc layer. We use SHT on Cars and the results are shown in Table 5. We observe that BN/WN may not help reduce the norm variance and the added BN does harm to SEC.

Reference: [1] Song et al. Deep Metric Learning via Lifted Structured Feature Embedding. CVPR 2016. [2] Yu et al. Deep Metric Learning with Tuplet Margin Loss. ICCV 2019. [3] Wang et al. Deep Metric Learning with Angular Loss. ICCV 2017. [4] Zhang et al. Heated-Up Softmax Embedding.