We thank all the reviewers (**R1**,**R2**,**R3**) for their feedback and suggestions.

Multi-task loss balancing (R1,R2). We have performed loss balancing with five different weights t in the multi-task loss $\mathcal{L}_m = t \cdot \mathcal{L}_c + (1-t) \cdot \mathcal{L}_r$ for the classification and regression losses. The results on OmniArt are reported in Table A. Our proposal is robust to the weight value, tuning the task weight is not vital. We obtain a moderate gain for both classification and regression with a weight of $t = 0.25$. For the multi-task baseline, emphasizing regression reduces the regression error, as the gradient magnitude of the regression loss is much lower than the one for the classification loss. However, this is not paired with an increase in classification accuracy. Across all weights, our proposal is preferred over the baseline. We will add the suggested experiment to Section 3.3. Thank you.

Table A: Multi-task comparison across task weights.

Task weight	0.01	0.10	0.25	0.50	0.75	0.90
Creation year (MAE ↓)						
MTL baseline	262.7	344.5	348.5	354.7	356.3	352.3
This paper	65.2	64.6	**64.1**	68.3	77.5	83.6
Art style (acc ↑)						
MTL baseline	44.6	47.9	49.5	47.2	47.7	47.1
This paper	46.6	51.2	**54.5**	52.6	52.5	51.4

Softmax cross-entropy loss (R3). We tried the suggested softmax cross-entropy loss on the hyperspherical similarities to all class prototypes. On CIFAR-100, we obtain an accuracy of 55.5 ± 0.2, compared to 65.0 ± 0.3 with our loss and 64.4 ± 0.4 with standard softmax cross-entropy, all with the same output dimensions and network settings. The softmax cross-entropy variant obtains a lower accuracy and is computationally more expensive, as it needs a similarity to all classes to compute the loss. The proposed loss requires a similarity to only one class during training. We will add the discussion to Section 3.1.

Hyperspherical separation and effect on classification (R1,R2). To **R2**, the statement on L95 regarding optimal separation is based on [33]. We will add the reference. Obtaining hyperspherical prototypes is fast: with 100 classes in 100 dimensions and 1,000 epochs, optimization takes only 4 seconds on a single 1080TI. Following **R1**'s suggestion, we have quantified the relation between separation and classification accuracy for the two hyperspherical baselines and our proposal with 100 output dimensions on CIFAR-100. We calculate the min (cosine distance of closest pair), mean (average pair-wise cosine distance), and max (cosine distance of furthest pair) separation. The results are shown in Table B. Our proposal obtains the highest maximum separation, indicating the importance of pushing many classes beyond orthogonality. One-hot prototypes do not push beyond orthogonality, while word2vec prototypes have a low minimum separation, which induces class confusion. To **R1**, we find degenerate cases on CIFAR-100 with three dimensions; the prototypes of multiple classes are overlapping. This case can either be solved by incorporating privileged information, or by using (at least two) more dimensions for the hypersphere (Table 2 of paper). We will add the discussions to Section 3.1.

Table B: Separation vs. classification.

	Separation ↑			acc.
	min	mean	max	
One-hot	**1.00**	1.00	1.00	62.1
Word2vec	0.26	1.01	1.32	57.6
This paper	0.95	**1.01**	**1.39**	**65.0**

More dimensions than classes (R1,R3). We have performed an additional experiment on CIFAR-100 with 200 output dimensions. We obtain an accuracy of 63.7 ± 0.4 (without privileged information) and 64.7 ± 0.1 (with privileged information), compared to 65.0 ± 0.3 for 100 dimensions. As foreseen by **R3**, the additional dimensions makes optimization more difficult, but privileged information alleviates this problem. We will insert the experiments with additional dimensions in Tables 1 and 2 and incorporate the discussion.

Extension to few-shot and open set classification (R2). In standard few-shot, the total number of classes is known [35], allowing us to place all classes on the hypersphere *a priori*. For open set classification, we can avoid retraining from scratch by extending the prototype separation. For a new class, we place it on the hypersphere (e.g. with privileged information) and then push all prototypes near the new one away. We will add these possibilities to the conclusions.

Clarifying one-hot and word2vec baselines (R1). For the word2vec and one-hot baselines (Tables 1 and 2), we only replace the prototypes, the loss and optimization remains the same. We will remove L184-185 to avoid confusion on the one-hot baseline. The word2vec baseline relies on word vectors of class names as the prototypes, which is feasible because the word vectors use the cosine similarity as distance metric [35], akin to hyperspherical prototype networks. The comparison to standard classification with softmax cross-entropy (Table 3) uses the same settings and dimensions, only the loss is altered. We believe this setup allows for a direct and fair comparison, but the description of baselines and experiments could surely be improved. We will do so and align the dimensions across Tables 1-3 to allow for cross-experiment comparisons.

We thank **R1** for the references. Compared to uniform sampling, we explicitly enforce maximum separation, because uniform sampling might randomly place prototypes near each other, which negatively affects the classification (Table B). We will cite and discuss the papers in Section 2.1. To **R2**, the sharp spike in Figure 4 is due to a learning rate decay after 100 epochs. We will update the caption of Figure 1b to increase clarity and remove the typo. We thank the reviewers.