Model	# of params	#of clusters	CUB-SS	CUB-PS	AwA-SS	AwA1-PS	FLO	Average
LDF(2018) [1]	426.4M	-	67.1	67.5	83.4	65.5	-	-
Resnet152	60.2M	-	66.9	67.3	81.0	67.5	64.0	69.3
Ours w/ SPN	61.0M	2	70.1	70.5	**83.7**	68.5	64.2	71.4
Ours	61.0M/42.5M	2	**70.5**/66.5	71.0/67.4	83.5/**82.9**	**68.8/66.1**	**65.9/65.6**	**71.8/69.7**
	-	3	69.2/**67.3**	**71.7**/67.1	82.4/82.6	66.3/66.5	65.8/64.7	71.1/69.4
	-	4	70.2/67.1	71.3/**67.6**	82.0/81.9	68.4/65.9	64.2/**65.6**	71.2/69.6

Table 1: Zero-shot learning results on three benchmarks. The number of parameters is calculated for the CUB dataset. We report the results of our model **without/with** sharing the CNN parameters for the input image and the local patches.

Method	CUB $A_{\mathcal{U}\to\mathcal{T}}$	CUB $A_{\mathcal{S}\to\mathcal{T}}$	CUB H	AwA1 $A_{\mathcal{U}\to\mathcal{T}}$	AwA1 $A_{\mathcal{S}\to\mathcal{T}}$	AwA1 H	AwA2 $A_{\mathcal{U}\to\mathcal{T}}$	AwA2 $A_{\mathcal{S}\to\mathcal{T}}$	AwA2 H	SUN $A_{\mathcal{U}\to\mathcal{T}}$	SUN $A_{\mathcal{S}\to\mathcal{T}}$	SUN H
DEM [2]	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1	20.5	34.3	25.6
RN* [3]	**38.1**	61.1	47.0	31.4	**91.3**	46.7	30.0	**93.4**	45.3	20.1	35.6	25.7
LDF* [1]	26.4	**81.6**	39.9	9.8	87.4	17.6	-	-	-	-	-	-
Ours*	36.7	71.3	**48.5**	**37.6**	87.1	**52.5**	**36.0**	84.3	**50.5**	**22.3**	**39.5**	**28.5**

Table 2: Generalized zero-shot learning results (%). H denotes the harmonic mean. * means end-to-end training.

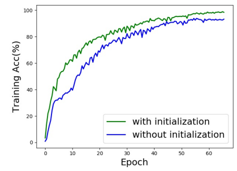

Figure 1: Training curve.

We first thank all reviewers for the valuable feedback.

Reviewer1

Q1: **Much more parameters.** To prove that the gain of performance is not totally from the higher capacity network, we conduct experiments using Resnet152 as the backbone with end-to-end finetune, which has a comparable amount of parameters to ours. As shown in Table 1, our model outperforms Resnet152 by 3.6%(71.8% v.s. 69.3%). Besides, our model has **significantly less parameters** than the best competing model LDF [1] while performing much better. We also can reduce the number of parameters by using the same CNN for the image and the part patches as you suggested, but the ZSL performance is slightly degraded roughtly from 71% to 69% as shown in Table 1 (separated with slashes).

Q2: The importance of weights initialization. We present the training curves of our model with/without weights initialization in Fig 1. We see that initializing the attention layers speeds up the learning and finally achieves a greater accuracy. We will add more detailed analysis in the final version of the paper.

Q3: The number of clusters. As shown in Table 1, we increase the number of clusters to 4 and find little performance improvement. Besides, we observe more maps introduce the attention redundancy, i.e. maps attend to the same region.

Q4: Results for generalized ZSL. As shown in Table 2, our model outperforms the other SOTA models (based on H).

Other comments. The CNN pretrained to provide psuedo labels for clustering is the same backbone used in our model, otherwise it would give erroneous peak as you agreed. We will cite the relevant papers you suggest.

Reviewer2

Q1: About Cropping network. In fact, to obtain better representation for finer localized cropped region x_i^{part}, our method also utilizes the bilinear sampling to adaptively zoom the cropped region x_i^{part} to the same size with the original image. Concretely, for a point (i, j) of the zoomed region, its value $x_{(i,j)}^{zoom}$ can be computed bilinearly combining the values of nearest four points in the cropped region. Formally, $x_{(i,j)}^{zoom} = \sum_{\alpha,\beta} |1-\alpha-\{i/\lambda\}||1-\beta-\{j/\lambda\}|x_{(m,n)}^{part}$, where $m = [i/\lambda]+\alpha+z_x-z_s, n = [j/\lambda]+\beta+z_y-z_s$, α = 0 or 1, β = 0 or 1, λ is the upsampling factor, $\lambda = t/t_s$ (t is the size of the original image) and $[\cdot]$ and $\{\cdot\}$ is the integral and fractional part, respectively. We will add the detailed description in the final version of the paper. Spatial Transformer Network(SPN) is an alternative of our cropping net. When replacing it with SPN, we find the performance changes little as shown in Table 1.

Q2: About triplet loss. We agree that the normalization will change the relative distance of two points. There is a typo leading to misunderstanding in the paper. We actually use the normalized version of ϕ in the embedding softmax loss so that only normalized features are considered and used in training and inference phases. Please refer to [4]. We will add more discussion in the final version of paper. **Other comments:** The local patches are going through the same CNN while the input image is going through a different CNN. We will mark it in the figure.

Reviewer3

The competing method LDF [1] used a single attention scheme and we have shown our superiority to it in both the model design and the performance. We will add more explanation about how the extracted features are used and add a reference to the appendix. Please kindly refer to our response to other reviewers.

References

[1] Li et al. Discriminative learning of latent features for zero-shot recognition. In *CVPR*, pages 7463–7471, 2018.

[2] Zhang et al. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 2021–2030, 2017.

[3] Sung et al. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018.

[4] Wang et al. Normface: l2 hypersphere embedding for face verification. In *ACMMM*, pages 1041–1049. ACM, 2017.